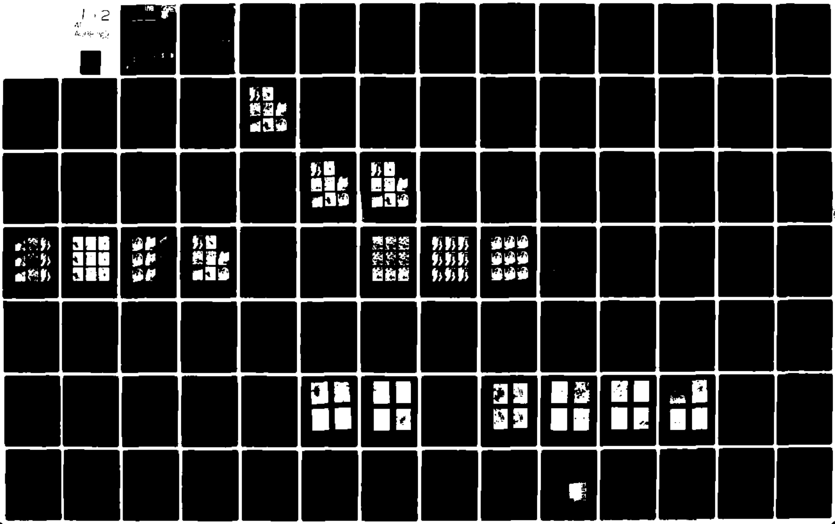
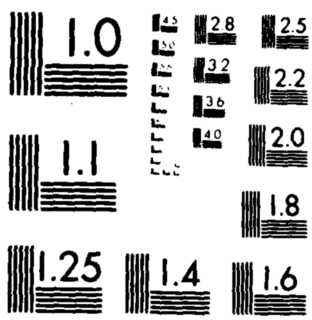
AD-A088 763 CONTROL DATA CORP MINNEAPOLIS MN IMAGE SYSTEMS DIV F/G 5/8
MAPS IMAGE COMPRESSION.(U)
MAY 80 A E LABONTE, T E ROSENTHAL F30602-78-C-0253
UNCLASSIFIED ISD-78003-80-R1 RADC-TR-80-173 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AD A088763

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-80-173
2. GOVT ACCESSION NO.: AD-A088 763
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): MAPS IMAGE COMPRESSION
5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report, 13 Sep 78 - 18 Jan 80
6. PERFORMING ORG. REPORT NUMBER: ISD-78003-80-R1
7. AUTHOR(s): Anton E. LaBonte, Tom E. Rosenthal
8. CONTRACT OR GRANT NUMBER(s): F30602-78-C-0253
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Control Data Corporation, Image System Division, Minneapolis MN 55440
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F, 45941803
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRRE), Griffiss AFB NY 13441
12. REPORT DATE: May 1980
13. NUMBER OF PAGES: 110
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report): Same
18. SUPPLEMENTARY NOTES: RADC Project Engineers: Douglas J. Praska, 1Lt, USAF (IRRE); Robert LaSalle (IRRP)
19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
   Adaptive Decompression
   Error Detection
   Macro Fidelity Control
   Contrast Control
20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

~~The work documented in~~ this report encompasses several areas of technique refinement regarding the MAPS image compression technique which were identified in an earlier RADC final technical report, RADC-TR-77-405 dated December 1977 (A050679). These areas include: adaptive decompression, contrast control simplification, resolution code error detection and recovery, and interactive Macro-fidelity control.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

S/C 408732

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

The following document constitutes the technical final report on Contract F30602-78-C-0253, MAPS IMAGE COMPRESSION. The work was performed in the Information Research Department, Information Sciences Division, Control Data Corporation by Dr. A.E. LaBonte, principal investigator, and Mr. T.E. Rosenthal, microcode development and implementation.

Thanks are extended to Mr. R. LaSalle and Lt. D. Praska of the Rome Air Development Center who shared the role of project technical monitoring. Their support, enthusiasm and guidance throughout the effort are sincerely appreciated.

Accession For
NTIS GRA&I ☒
DDC TAB ☐
Unannounced ☐
Justification________

By________
Distribution/
Availability Codes

Dist.	Avail and/or special
A	

DTIC
ELECTE
SEP 3 1980
S D
D

ORIGINAL CONTAINS COLOR PLATES: ALL DDC
REPRODUCTIONS WILL BE IN BLACK AND WHITE.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

LIST OF FIGURES

LIST OF FIGURES (Cont.)

LIST OF TABLES

EVALUATION

This effort has significantly advanced and refined a very unique and powerful image compression technique known as Micro-Adaptive-Picture-Sequencing (MAPS). It has brought the Air Force one step closer to the realization of compressing intelligence imagery with excellent results. This technique when incorporated within the design and production of imagery exploitation, storage and retrieval, and distribution systems will render the Air Force major cost savings due to reductions in hardware. The effort has contributed significantly towards accomplishing goals of Technical Program Objective (TPO) R2C.

Douglas J Praska

DOUGLAS J. PRASKA, 1Lt, USAF
Project Engineer

SECTION ONE

SUMMARY

MICRO-ADAPTIVE PICTURE SEQUENCING (MAPS) is a digital image data compression technique which originated at Control Data Corporation and underwent extensive exploration and elaboration with sponsorship from Rome Air Development Center under contract F30602-76-C-0350. The effort and results described herein represent the next major phase in which several further MAPS developments and evaluations are covered.

1.1 OBJECTIVES

Five areas of MAPS are investigated:

- Adaptive Decompression,
- Simplified Micro-Fidelity Specification,
- Error Propagation Protection,
- Interactive Macro-Fidelity Specification,
- Image Processing System Implementation.

The objective of the adaptive decompression task is to develop and refine an efficient approach to removal of the "blockiness" artifacts present in the MAPS block decompressed imagery. A variable "convolution" which adapts on the basis of the local pattern of element sizes and intensities is explored with a variety of weighting functions. Residual gray-scale contouring is masked using a small-amplitude random "dither".

The objective of the simplified micro-fidelity specification task is to reduce the number of degrees of freedom in assigning the contrast control matrix and to develop guidelines for selection of those parameters which remain. The simplification is to be achieved without sacrifice of significant fidelity at a given compression level.

The objective of the error propagation protection task is to assess and optimize the error detection and recovery potential inherent in the residual redundancy of the MAPS resolution codes. Particular attention is applied to protection from "catastrophic" errors in which whole subframes are lost or inserted destroying the large-scale integrity of the image.

The objective of the interactive macro-fidelity specification task is to explore the potential for describing arbitrary geometric boundaries between regions with different fidelity requirements. Exploitation of the "transparency" of the MAPS decompression process to variation in the fidelity control on compression is exploited. The implementation is to be interactive and compatible with the on-line demonstration of the final task.

The objective of the image processing system implementation task is to demonstrate the MAPS technique on a very high speed processor. Generic coding considerations for microcode implementations are emphasized in this development.

1.2 BASIC MAPS CONCEPTS — A BRIEF REVIEW

MAPS is basically a contrast-adaptive variable-resolution image encoder. Detailed description of the technique is given in the final report from the prior contract*. Here, the two central concepts of contrast adaptation and constrained local resolution variation are briefly reviewed.

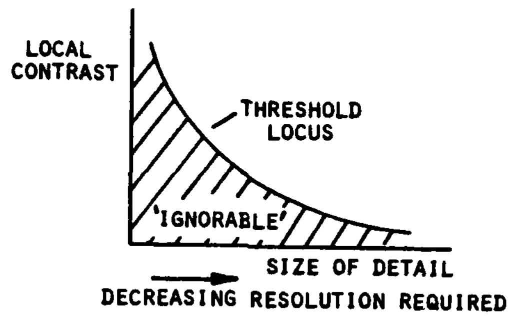
*RADC-TR-77-405, IMAGE COMPRESSION TECHNIQUES, A.E. LaBonte and C.J. McCallum, Control Data Corporation, December 1977. (A050679)

The basis of MAPS control is the simple "vision heuristic" illustrated in Figure 1-1. In standard PCM matrix or raster image coding, the pixel size is uniform throughout the image and the resolution is fixed by the smallest detail to be displayed. In typical imagery, this resolution is required only in very localized portions of the image and the remainder can be adequately characterized by much coarser elements. In essence, the heuristic asserts that the highest resolution is required only at points where the detail exhibits strong contrast. More extended structures, however, are perceived at much lower contrast. Thus, there is a threshold locus in contrast-size space below which detailed structure is "ignorable" and resolution may be reduced until this threshold is exceeded.

The basis of MAPS coding is the variable resolution constraint illustrated in Figure 1-2. Image element size is allowed to grow only in powers of four (in terms of original pixel count) and only within natural binary boundaries. Element size is denoted by a resolution or level code which has the form $\log_4$ (original pixel count) and is given explicitly for each element along with its intensity assignment. Note that restriction to elements commensurate with the natural binary boundaries allows the element position to remain implicit; it is implied by the sequence of resolution codes which must fit together in a directed exhaustive covering of the region. Observe also that the $\log_4$ integer coding provides both a very compact code and a large dynamic range. For the maximum element size of 16x16 used here, only five level codes (L=0, 1, 2, 3, 4) are required to cover a range of 256:1 (4^L= 1, 4, 16, 64, 256).

The two concepts illustrated by Figures 1-1 and 1-2 work together as follows:

- Elements are always considered together in groups of four which are the subelements of a potential element at the next higher level (that is, they satisfy the implicit sequence constraint):



VISION HEURISTIC

Figure 1-1. Basis of MAPS: Adaptive Contrast Control

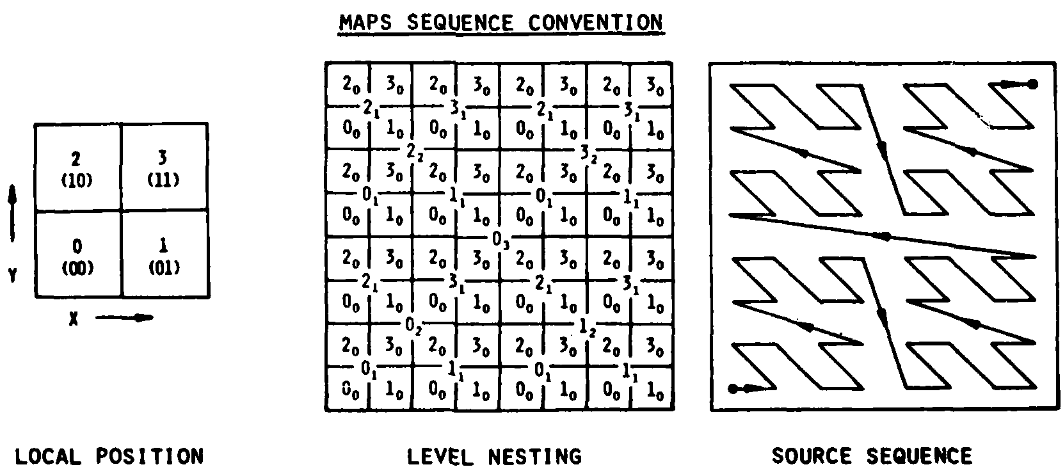


Figure 1-2. Basis of MAPS: Variable Resolution

- Contrast measures are defined as intensity differences among the four elements;
- Contrast thresholds are chosen on the basis of the level or resolution of the current elements and the contrast measures are tested against them;
- If any threshold is exceeded, the four elements are left at that level; if no threshold is violated, the four elements are combined to a single element at the next higher level and a composite intensity (the mean of the four subelements) is assigned;
- The process requires that all four subelements be available before transition to the next level is attempted;
- The process proceeds through the source image in a sequence consistent with the implicit position constraints (see Figure 2-1, shown up through an 8x8 or level 3 block);
- Element composition up to the maximum level value — the MAPS subframe size — is allowed.

The MAPS process thus results in a highly-nonlinear adaptation to the image scene content and a corresponding compact localized coding.

1.3 TEST IMAGERY

The further MAPS developments and evaluations undertaken here require a diverse set of test imagery to insure that the process is not inadvertently "tailored" to a specific image class. The image test set is displayed in Figure 1-3. The samples were chosen to represent a wide range of image scale, image type (photo, radar, annotated line image), scene content (natural versus cultural), and contrast.

Image size was chosen as 624 pixels/line by 480 lines to insure compatibility with the display system in the high speed image processing system. The intensities in the original imagery were encoded to six bits.

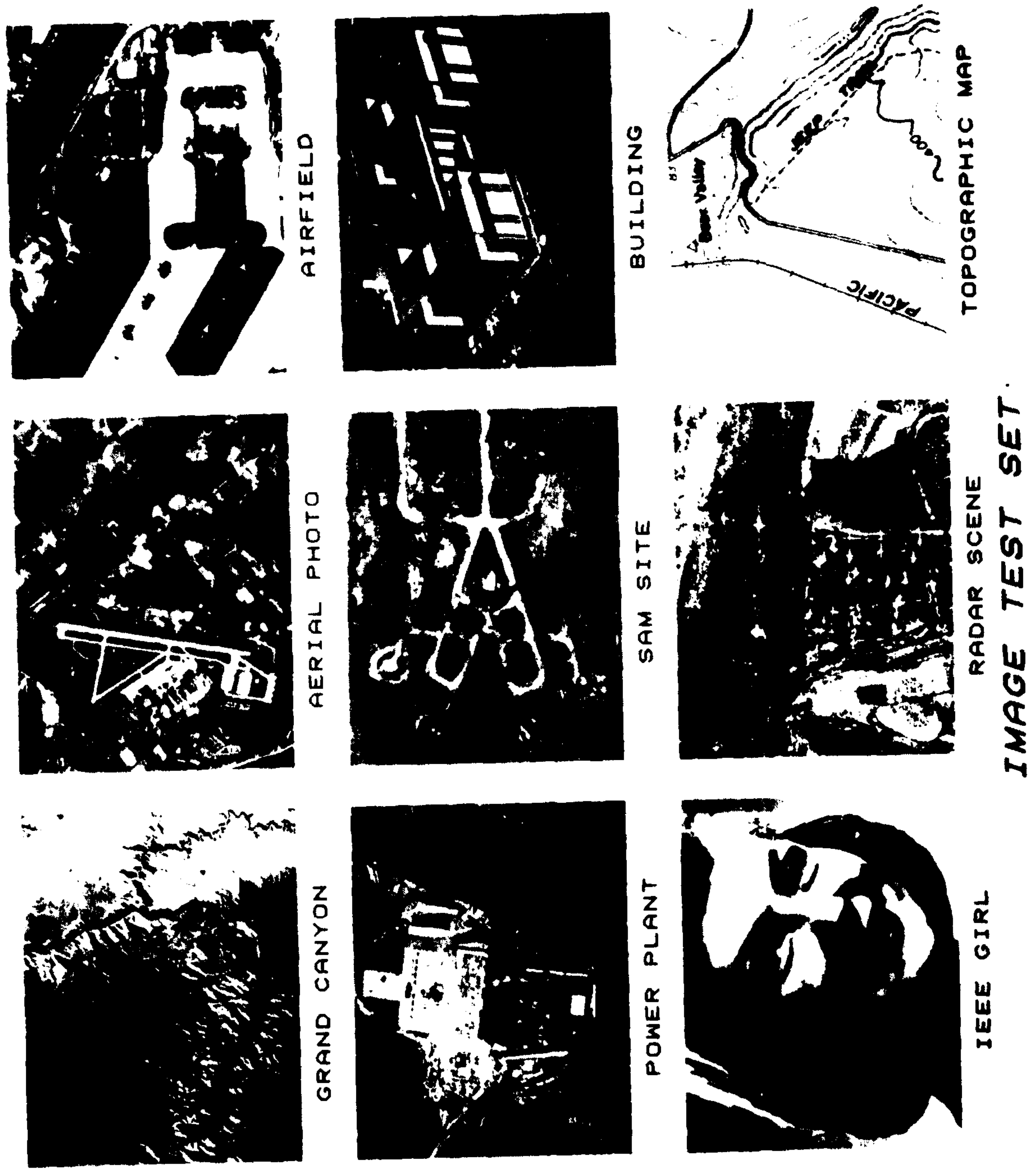


Figure 1-3. Test Imagery Set

The MAPS process is clearly sensitive to contrast but not to the mean gray scale in the image. Thus, all images (except the topographic map) were adjusted to approximately the same mean to ease hardcopy dynamic range requirements and to make the denominators in mean square error evaluations more comparable from image to image. The mean and standard deviation (on a scale of 0 → 63) for each image before and after the scale shift are listed in Table 1-1. Note that the shift did not result in any significant reduction in the standard deviation due to forced saturation.

1.4 RESULTS AND ORGANIZATION OF REPORT

The remainder of the report deals with each of the task areas in turn. A brief summary of the key results in each area is given below.

Results for the adaptive decompression effort are described in Section Two. The process was updated to include an efficient table-driven version of the "convolution" which permits an arbitrary set of weighting functions to be included. Four cases were compared with the basic block decompression mode — uniform, pyramid, "Markov", and Gaussian weighting. In all cases, the adaptive decompression improved the mean square error (MSE) over that for block decompression. The average improvement was just over 11% of the block MSE for compressions at about one bit per pixel. The weighting function for best improvement varied among the nine test images but the Gaussian form held a slight overall advantage. In any event, the range from least to best improvement over the weighting types was only about one percent of the block decompression MSE.

Results for the micro-fidelity specification simplification effort are described in Section Three. Direct specification for the sixteen elements of the matrix was replaced by a matrix generation procedure in terms of four parameters. Experiments with the test image set, in turn, demonstrate that three of the four can be assigned "universal" values with little loss in fidelity performance at fixed compression. Moreover, the range of choice of these universal values is quite wide giving credence to the assertion that they will be applicable to imagery beyond that represented by the

TABLE 1-1. TEST IMAGE STATISTICS

IMAGE	SOURCE DATA		REVISED TEST SET	
	MEAN	STANDARD DEVIATION	MEAN	STANDARD DEVIATION
GRAND CANYON	35.4	15.32	26.4	15.30
AERIAL PHOTO	17.5	7.20	22.5	7.20
AIRFIELD	31.9	18.85	26.0	18.81
POWER PLANT	42.6	8.57	26.6	8.50
SAM SITE	18.1	8.84	24.1	8.83
BUILDING	50.7	7.32	24.7	7.32
IEEE GIRL	33.2	15.36	26.2	15.34
RADAR SCENE	21.3	8.59	21.3	8.59
TOPOGRAPHIC MAP	9.9	6.86	9.9	6.86

test set. Thus, control is simplified to selection of a single quantity — the overall contrast scale. Although this remaining parameter appears to be strongly image dependent, one "rule of thumb" for its selection did emerge: Given the compression at one contrast scale value, the compression at another scale is approximately $C_2 = C_1 (T_2/T_1)^a$. Here C is the compression in bits per pixel, T is the contrast threshold scale, and exponent a depends somewhat on the "universal" values chosen for the other generator parameters.

The error propagation protection effort is described in Section Four. The resolution codes are revised so that the last element in each MAPS subframe is designated by a termination code. The code space is then assigned such that any single-bit error in a termination code is immediately detected as an illegal code and, correspondingly, no other legitimate level code can be converted into a false termination code via a single-bit error. This process is then strengthened by invoking a universal constraint which must be satisfied by the element count between termination codes. Together these processes imply that a net of three undetected false terminators must be added within one subframe or a string of three successive legitimate terminators must be deleted before a catastrophic subframe position error will arise. Note that this entire process is achieved at no change in the basic MAPS compression level. Beyond this level of catastrophic error protection, the local error detection potential of the inherent MAPS resolution consistency relations is formulated in terms of an allowed-level constraint map. Finally, guidelines for data reorganization are outlined for cases where more elaborate external error control is to be added.

Results for the interactive macro-fidelity specification and image processing system implementation efforts are highly intertwined and are covered together in Section Five. Descriptions of the image processing system, the MAPS microcode implementation, and the user operating procedures are outlined in the text and elaborated in Appendices A, B, and C. Although

the dynamics of the process cannot be captured, several examples in the form of direct photographs of the video screen are included. The speed enhancement of the image processing system implementation relative to MAPS on a large-scale high-speed general purpose computer is observed to be about 20:1!

Finally, the MAPS technique has reached a level of maturity which suggests the preparation of a "transportable software" package written in a high-level language, probably FORTRAN. Section Six concludes the report with recommendations for the user options which such a package should contain.

SECTION TWO
ADAPTIVE DECOMPRESSION

The potential for adaptive decompression of the MAPS data stream is based on the additional information implied by the MAPS resolution codes. Two concepts are central to this adaptation. First, for the region within a MAPS element, the element size (keyed explicitly by the resolution code) gives a measure of the "correlation length" in that region. This, in turn implies an appropriate size for an adaptive convolution window to be used in resampling the image. Second, the relative size of the MAPS elements (again keyed by the resolution codes) gives a direct measure of the image "activity level" in the area. Small elements encode significant local structure and should not be included in the resampling used to smooth nearby larger elements.

2.1 Adaptive "Convolution" Process

In view of the observations above, adaptive decompression mode based on a resampling and smoothing convolution is a natural adjunct to MAPS. Four major dimensions to the adaptivity have been identified. They are:

- Convolution window size,
- Element-size resampling thresholds,
- Convolution weighting functions, and
- "Dither" selection.

Adaptation for each of these is a function of the MAPS resolution code for the target element and its surrounding elements.

The geometry for the process is exhibited in Figure 2-1. Based on the correlation length argument, the convolution window size should be chosen to approximate the size of the target element. In addition, it should be symmetric about the target pixel. A window edge which is chosen to be one pixel less than the target block edge satisfies these conditions. This yields window sizes of 1x1, 3x3, 7x7, and 15x15 for levels one through four respectively.

Note that restriction to a window size no larger than the target block has another significant advantage. As seen in Figure 2-1, the target pixels in each quadrant interact with only part of the surround:

Target pixel quadrant	Active regions in window
I	0 1 2 3 5 7
II	0 2 3 4 6 8
III	0 5 7 9 10 11
IV	0 6 8 10 11 12

Moreover, element-size resampling thresholds apply to the MAPS elements surrounding the target element and operate on the relative difference between target and surround resolution codes. If only elements no smaller than one level below the target or central block are actively used, then each of the twelve surround regions has a uniform MAPS input intensity. As a consequence of the window size and activation restrictions, each convolution will have at most six independent MAPS inputs. This gives significant potential for improving the convolution efficiency.

The activation restriction is also motivated by the scene structure. The presence of MAPS elements smaller than one level below the central block is a priori evidence of more detailed and localized image structure which should not be permitted to contaminate the target pixel.

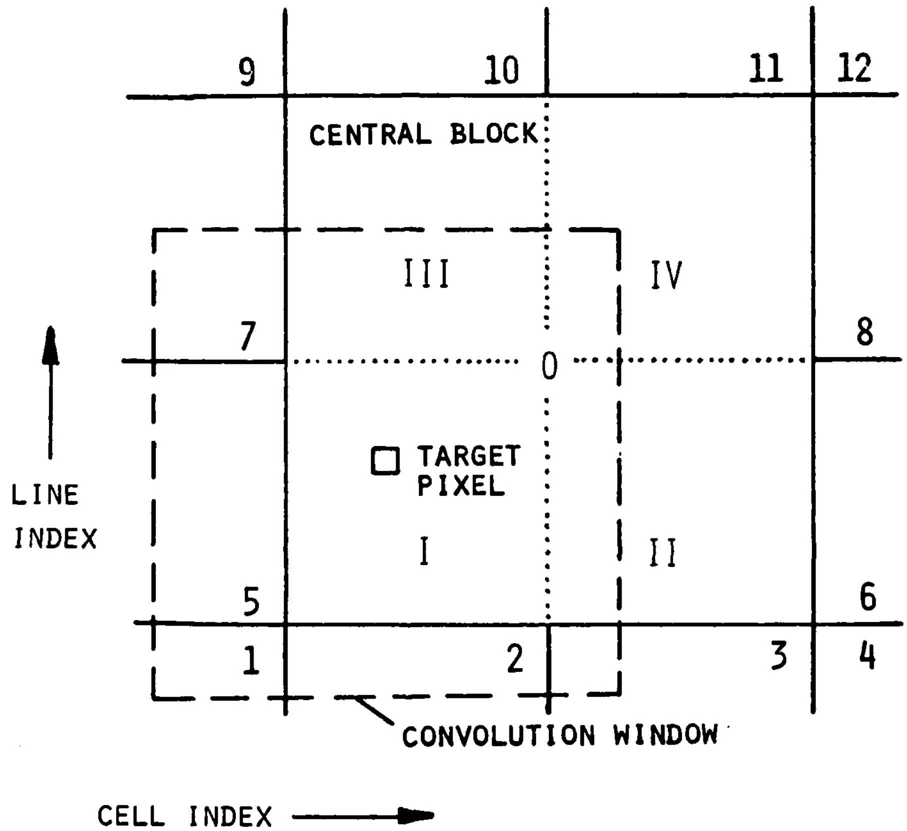


Figure 2-1. Adaptive Decompression Geometry

The adaptive "convolution" (a conflict in terms since the window and activation adaptation destroy spatial invariance) is implemented as a table-driven process. Since each convolution depends on only six independent quantities, the weights can be presummed for each target pixel location and stored as six-component vectors addressed by the target position. For the 15x15 window, for example, the convolution is reduced from 225 multiplies and adds to 6 multiplies, adds, and activation level comparisons — a dramatic improvement in implementation. In this table-driven form, an arbitrary weighting function over the window can be supplied to the presum operation. From there on, the process is independent of the weight form.

More detailed descriptions of the weighting functions and dither addition are covered in the next two subsections.

2.2 WEIGHTING FUNCTIONS

Four different weight functions — uniform, pyramid, "Markov," and Gaussian — were used.

2.2.1 Uniform

With uniform weighting, the target pixel depends equally on all active pixels within the window. The input weighting functions for levels two, three, and four are shown in Figure 2-2 for the uniform case.

2.2.2 Pyramid

With pyramid weighting, the contribution of each active pixel decreases linearly with the larger of its x or y separations from the target pixel. The pyramid input weight function is displayed in Figure 2-3.

WEIGHTING: UNIFORM

LEVEL 2

1.000	1.000	1.000
1.000	1.000	1.000
1.000	1.000	1.000

LEVEL 3

1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000

LEVEL 4

1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Figure 2-2. Uniform Weighting Function

WEIGHTING: PYRAMID

LEVEL 2

1.000	1.000	1.000
1.000	2.000	1.000
1.000	1.000	1.000

LEVEL 3

1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	2.000	2.000	2.000	2.000	2.000	1.000
1.000	2.000	3.000	3.000	3.000	2.000	1.000
1.000	2.000	3.000	4.000	3.000	2.000	1.000
1.000	2.000	3.000	3.000	3.000	2.000	1.000
1.000	2.000	2.000	2.000	2.000	2.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000

LEVEL 4

1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	1.000
1.000	2.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	2.000	1.000
1.000	2.000	3.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	3.000	2.000	1.000
1.000	2.000	3.000	4.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000	4.000	3.000	2.000	1.000
1.000	2.000	3.000	4.000	5.000	6.000	6.000	6.000	6.000	6.000	5.000	4.000	3.000	2.000	1.000
1.000	2.000	3.000	4.000	5.000	6.000	7.000	7.000	7.000	6.000	5.000	4.000	3.000	2.000	1.000
1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000	7.000	6.000	5.000	4.000	3.000	2.000	1.000
1.000	2.000	3.000	4.000	5.000	6.000	7.000	7.000	7.000	6.000	5.000	4.000	3.000	2.000	1.000
1.000	2.000	3.000	4.000	5.000	6.000	6.000	6.000	6.000	6.000	5.000	4.000	3.000	2.000	1.000
1.000	2.000	3.000	4.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000	4.000	3.000	2.000	1.000
1.000	2.000	3.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	3.000	2.000	1.000
1.000	2.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	3.000	2.000	1.000
1.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	2.000	1.000
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Figure 2-3. Pyramid Weighting Function

2.2.3 "Markov"

Tonal imagery is often modelled as a first order Markov process with a high correlation coefficient between adjacent pixels. A weighting function in which each contribution is weighted by p^{δ} where p is the correlation between adjacent pixels and δ is the distance from the target pixel (in units of the interpixel spacing) was also implemented. The weighting functions for $p = 0.9$ are shown in Figure 2-4.

2.2.4 Gaussian

Finally, a Gaussian weighting in which the center to corner distance is 2σ, was employed. Note here that σ varies with the local block size which is consistent with the concept that this size is a measure of the correlation length.. The Gaussian weighting functions are given in Figure 2-5.

2.3 DITHER

With the convolution process, the blockiness of the image is removed but the discrete quantization of the intensities still leaves a residual gray-scale contour pattern. The addition of dither is largely a cosmetic process to mask this contouring and the amplitude must be sufficient to cover the boundaries but small enough not to blur or obliterate real image features.

The effectiveness of the dither is critically dependent on the point in the process at which it is added. The convolution process will typically yield a non-integer value for the smoothed intensity and the dither should be added to this value <u>prior</u> to requantization to the final gray scale value. In this manner, the <u>density</u> of pixels of each quantized level will shade gradually from one gray scale to the next. In this case a dither amplitude of two gray scale levels is typically sufficient.

WEIGHTING: "MARKOV"

LEVEL 2

.862	.900	.862
.900	1.000	.900
.862	.900	.862

LEVEL 3

.640	.684	.717	.729	.717	.684	.640
.684	.742	.790	.810	.790	.742	.684
.717	.790	.862	.900	.862	.790	.717
.729	.810	.900	1.000	.900	.810	.729
.717	.790	.862	.900	.862	.790	.717
.684	.742	.790	.810	.790	.742	.684
.640	.684	.717	.729	.717	.684	.640

LEVEL 4

.352	.379	.404	.428	.448	.464	.475	.478	.475	.464	.448	.428	.404	.379	.352
.379	.409	.439	.468	.493	.514	.527	.531	.527	.514	.493	.468	.439	.409	.379
.404	.439	.475	.509	.541	.567	.584	.590	.584	.567	.541	.509	.475	.439	.404
.428	.468	.509	.551	.590	.624	.648	.656	.648	.624	.590	.551	.509	.468	.428
.448	.493	.541	.590	.640	.684	.717	.729	.717	.684	.640	.590	.541	.493	.448
.464	.514	.567	.624	.684	.742	.790	.810	.790	.742	.684	.624	.567	.514	.464
.475	.527	.584	.648	.717	.790	.862	.900	.862	.790	.717	.648	.584	.527	.475
.478	.531	.590	.656	.729	.810	.900	1.000	.900	.810	.729	.656	.590	.531	.478
.475	.527	.584	.648	.717	.790	.862	.900	.862	.790	.717	.648	.584	.527	.475
.464	.514	.567	.624	.684	.742	.790	.810	.790	.742	.684	.624	.567	.514	.464
.448	.493	.541	.590	.640	.684	.717	.729	.717	.684	.640	.590	.541	.493	.448
.428	.468	.509	.551	.590	.624	.648	.656	.648	.624	.590	.551	.509	.468	.428
.404	.439	.475	.509	.541	.567	.584	.590	.584	.567	.541	.509	.475	.439	.404
.379	.409	.439	.468	.493	.514	.527	.531	.527	.514	.493	.468	.439	.409	.379
.352	.379	.404	.428	.448	.464	.475	.478	.475	.464	.448	.428	.404	.379	.352

Figure 2-4. "Markov" Weighting Function, Correlation=0.9

WEIGHTING: GAUSSIAN

LEVEL 2

.135	.368	.135
.368	1.000	.368
.135	.368	.135

LEVEL 3

.135	.236	.329	.368	.329	.236	.135
.236	.411	.574	.641	.574	.411	.236
.329	.574	.801	.895	.801	.574	.329
.368	.641	.895	1.000	.895	.641	.368
.329	.574	.801	.895	.801	.574	.329
.236	.411	.574	.641	.574	.411	.236
.135	.236	.329	.368	.329	.236	.135

LEVEL 4

.135	.176	.221	.265	.306	.339	.360	.368	.360	.339	.306	.265	.221	.176	.135
.176	.230	.288	.346	.399	.442	.470	.480	.470	.442	.399	.346	.288	.230	.176
.221	.288	.360	.433	.500	.553	.588	.600	.588	.553	.500	.433	.360	.288	.221
.265	.346	.433	.520	.600	.665	.707	.721	.707	.665	.600	.520	.433	.346	.265
.306	.399	.500	.600	.693	.767	.815	.832	.815	.767	.693	.600	.500	.399	.306
.339	.442	.553	.665	.767	.849	.903	.922	.903	.849	.767	.665	.553	.442	.339
.360	.470	.588	.707	.815	.903	.960	.980	.960	.903	.815	.707	.588	.470	.360
.368	.480	.600	.721	.832	.922	.980	1.000	.980	.922	.832	.721	.600	.480	.368
.360	.470	.588	.707	.815	.903	.960	.980	.960	.903	.815	.707	.588	.470	.360
.339	.442	.553	.665	.767	.849	.903	.922	.903	.849	.767	.665	.553	.442	.339
.306	.399	.500	.600	.693	.767	.815	.832	.815	.767	.693	.600	.500	.399	.306
.265	.346	.433	.520	.600	.665	.707	.721	.707	.665	.600	.520	.433	.346	.265
.221	.288	.360	.433	.500	.553	.588	.600	.588	.553	.500	.433	.360	.288	.221
.176	.230	.288	.346	.399	.442	.470	.480	.470	.442	.399	.346	.288	.230	.176
.135	.176	.221	.265	.306	.339	.360	.368	.360	.339	.306	.265	.221	.176	.135

Figure 2-5. Gaussian Weighting Function, 2σ in Corner

Each particular dither addition is simply drawn as the next output from a pseudo random number generation routine.

2.4 COMPARISONS AND EXAMPLES

The results of the adaptive decompression experiments are summarized in Table 2-1 which lists mean square error (MSE) values for each of the nine test images under each of the weighting schemes. Note that block decompression can be included under this formalism by interpreting the weighting pattern as a delta function. The mean square error is defined as:

$$MSE = \sum_{\ell=1}^{N_\ell} \sum_{k=1}^{N_k} \left[M(k,\ell) - I(k,\ell)\right]^2 \Bigg/ \sum_{\ell=1}^{N_\ell} \sum_{k=1}^{N_k} \left[I(k,\ell)\right]^2 , \qquad (2.4\text{-}1)$$

where I is the original image intensity, M is the corresponding MAPS decompressed intensity and the sums extend over all pixels in the image.

All compressions are about 6:1 or approximately one bit per pixel. Relative to the block decompression, the various adaptive decompressions all show improvement in the MSE values. The mean improvement is on the order of 11.5% of the block MSE and the spread among the various adaptive weightings is typically less than 1% of the block MSE. Thus, the overall effect of adaptive decompression is significant but the choice among weighting functions is much less so. The Gaussian shows a slight edge on seven of the nine images. Since the only difference involves the initial specification of the weights pattern, no computational cost is incurred by choosing the more complex function.

Finally, the effect of additive dither (with uniform weighting) is included in the last column of Table 2-1. Note that the MSE values with adaptation and dither are very similar to those for the original block

TABLE 2-1. MEAN SQUARE ERROR (MSE) IN PERCENT VERSUS ADAPTIVE CONVOLUTION WEIGHTING

Image	Compression Ratio	Weighting Type: Dither=0: Delta Block	Uniform	Pyramid	"Markov" Correlation 0.9	Gaussian	Uniform Dither=2
Grand Canyon	5.84	1.4664	1.3881	1.3800	1.3859	<u>1.3746</u>	1.4154
Aerial Photo	6.24	0.7718	0.6817	0.6760	0.6801	<u>0.6736</u>	0.7319
Airfield	5.96	0.2539	0.1941	<u>0.1929</u>	0.1936	0.1951	0.2205
Power Plant	5.99	0.4727	0.4194	0.4168	0.4186	<u>0.4157</u>	0.4589
Sam Site	6.38	0.3600	0.3323	0.3307	0.3318	<u>0.3300</u>	0.3702
Building	5.94	0.2999	0.2833	0.2788	0.2821	<u>0.2747</u>	0.3284
IEEE Girl	5.32	0.0849	<u>0.0717</u>	0.0721	0.0717	0.0735	0.1015
Radar Scene	5.92	1.0666	0.9733	0.9683	0.9721	<u>0.9656</u>	1.0244
Topo Map	6.14	1.3128	1.1539	1.1482	1.1511	<u>1.1447</u>	1.3820

decompression. Visual comparisons are provided by Figures 2-6 (block decompression) and 2-7 (adaptive decompression with dither). Suppression of blockiness is particularly noticeable in the GRAND CANYON, AERIAL PHOTO, SAM SITE, and TOPOGRAPHIC MAP images.

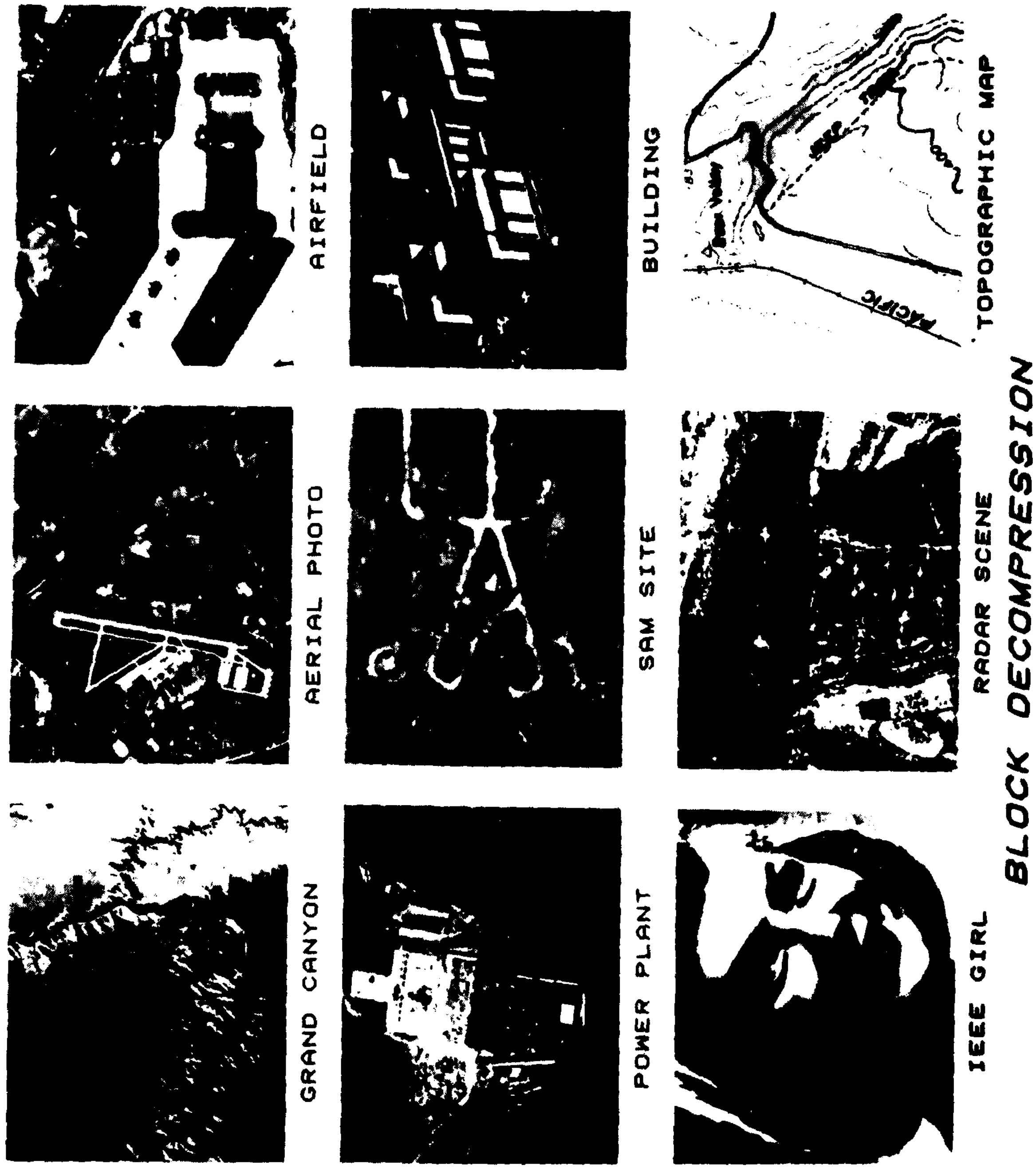


Figure 2-6. Block Decompression Examples, $C \approx 6:1$

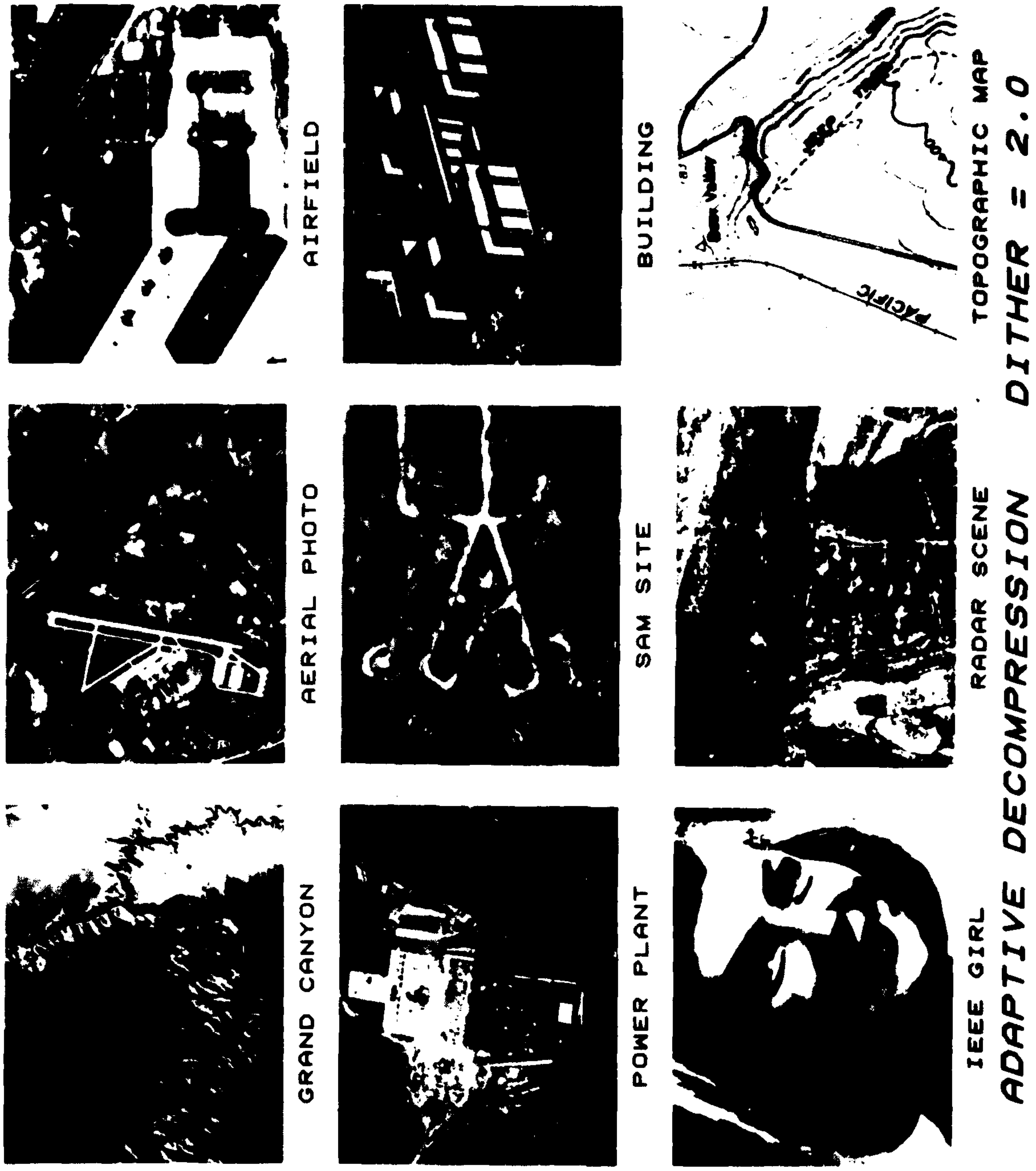


Figure 2-7. Adaptive Decompression Examples, C≈6:1; Uniform Weighting, Dither=2.

SECTION THREE
CONTRAST CONTROL SIMPLIFICATION

The basic MAPS control matrix for the selected subframe size of 16x16 original pixels involves four level-transition thresholds for each of four contrast types. Thus, each matrix requires specification of sixteen elements and a new matrix must be chosen each time the macro-fidelity requirement is changed. The goal of this task is to establish guidelines for selection of the matrix elements using fewer degrees of freedom.

3.1 THE CONTRAST CONTROL MATRIX

MAPS contrast control is illustrated by Figure 3-1. The control operation always deals with the four already-formed subelements of a potential MAPS element. The process proceeds as follows:

- The four intensities are sorted by increasing intensity;
- Four contrasts are defined as,

"extreme" contrast	-	the difference between the lowest and highest intensity,
"middle step" contrast	-	the difference between the two middle intensities,
"lower step" contrast	-	the difference between the two lowest intensities, and
"upper step" contrast	-	the difference between the two highest intensities;

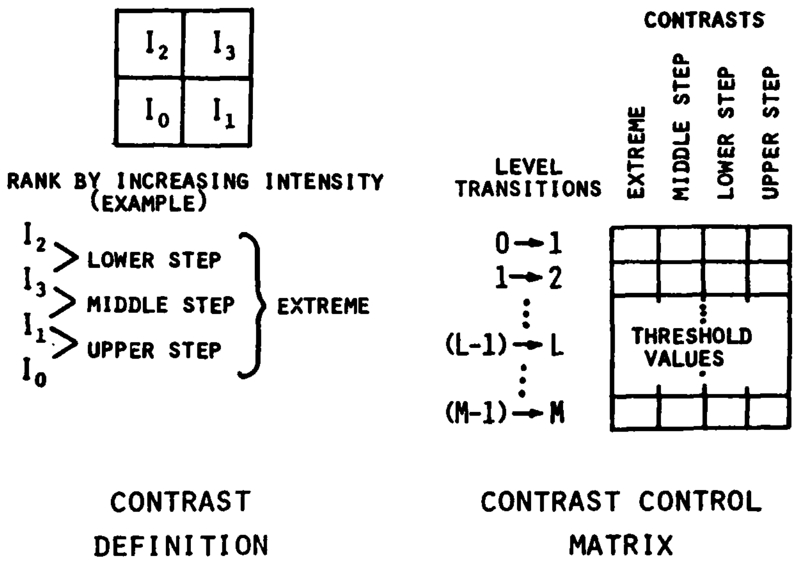


Figure 3-1. MAPS Contrast Control

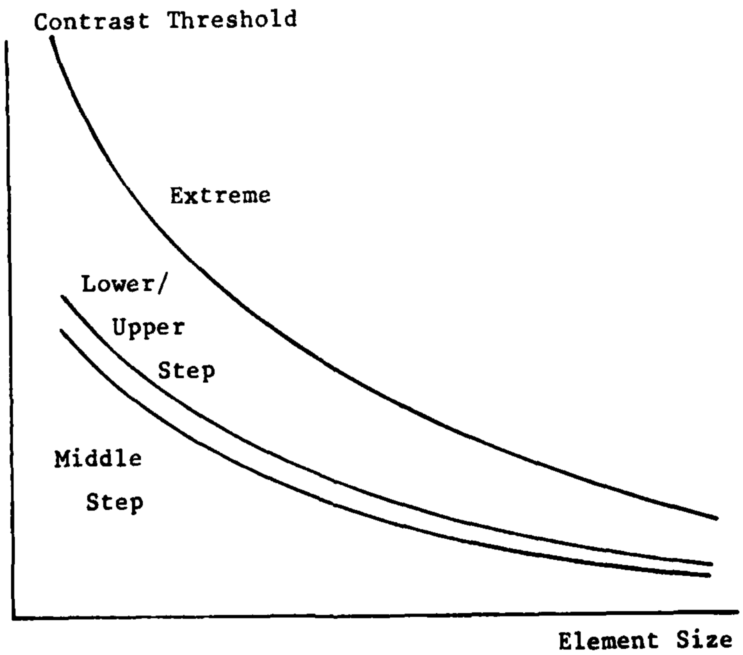


Figure 3-2. Contrast Thresholds

- Each contrast is tested against a corresponding threshold addressed by the contrast type and the level transition involved.

The qualitative behavior desired for the thresholds is illustrated in Figure 3-2. In general, the threshold decreases with increasing level transition or element size, and the "interior" contrast thresholds are smaller than the extreme threshold. Moreover, the middle step threshold is typically smaller than those for the lower and upper steps. (These conclusions were established in the previous MAPS contract efforts.)

3.2 GENERATOR PARAMETERS

The qualitative threshold behavior described above can be generated parametrically from a simple model involving four degrees of freedom.

The dimensionality reduction proceeds by partitioning the dependence on resolution code (the level transitions) and on contrast type into two independent factors. Empirical experience indicates that the resolution code dependence is one in which the transition thresholds fall exponentially with increasing MAPS level. If i denotes the resolution code (level transition from i to i+1) and B denotes the "taper" or base for the exponential decay, then this factor is B^{-i}.

The contrast type dependence requires that the various "step" contrasts - middle, lower, upper - are some fraction of the extreme contrast. Further, the "middle step" is biased to a slightly smaller than average fraction while the lower and upper step thresholds are equal and slightly larger than average. If F represents the fraction and Δ the bias, the middle step factor is $F(1-\Delta)$, relative to the extreme threshold and the upper and lower step factors are both $F(1+\Delta)$.

The only remaining parameter is an overall threshold scale factor, T_o. Combination of all these terms then yields

$$T_{Extreme} \quad (i \rightarrow i+1) \; = T_o \; B^{-i} \qquad i = 0, 1, 2, 3, \tag{3.2-1}$$

$$T_{Middle} \quad (i \rightarrow i+1) \; = T_o \; B^{-i} \; F(1-\Delta), \tag{3.2-2}$$

$$T_{Lower} \quad (i \rightarrow i+1) \; = T_o \; B^{-i} \; F(1+\Delta), \tag{3.2-3}$$

$$T_{Upper} \quad (i \rightarrow i+1) \; = T_o \; B^{-i} \; F(1+\Delta). \tag{3.2-4}$$

Equations (3.2-1) through (3.2-4) define the sixteen-element MAPS contrast control matrix in terms of four parameters: the taper, B, the step fraction, F, the step bias, Δ, and the threshold scale, T_o.

3.2.1 Exponential Taper Base Trade-Offs

Based on prior empirical experience, F and Δ were fixed initially at 0.5 and 0.1 respectively. Extensive exploration at five different values of the taper — B = 1.5, 2, 2.5, 3, 3.5 — was then undertaken. Several hundred compression runs varying the source image, the taper, and the contrast scale were made to establish the trade-off data base. The results were then analyzed in terms of the MSE by image at fixed compression levels. The MSE values for C = 0.6 bits/pixel are listed in Table 3-1 and plotted in Figure 3-3. From analyses of this type, the following observations were drawn:

- The optimum exponential taper base (B) for control matrix generation is essentially imagery independent.
- The optimum taper base is essentially compression independent (at least over 2.0 to 0.25 bits per pixel, the approximate range examined).

TABLE 3-1. Mean Square Error (MSE) as a Function of Control Matrix Taper Base (B) at Fixed Compression*, C = 0.6 bits/pixel.

Test Image	Taper Base, B 1.5	2.0	2.5	3.0	3.5
Grand Canyon	2.316%	2.155	2.147	2.183	2.242
Aerial Photo	1.167	1.064	1.048	1.058	1.063
Airfield	0.508	0.459	0.456	0.468	0.487
Power Plant	0.520	0.485	0.482	0.498	0.502
SAM Site	0.723	0.670	0.667	0.672	0.686
Building	0.533	0.495	0.494	0.510	0.527
IEEE Girl	0.162	0.150	0.146	0.150	0.153
Radar Scene	1.547	1.472	1.456	1.464	1.491
Topographic Map	2.882	2.691	2.765	2.991	3.314

* MSE values for C = 0.6 b/p were estimated through interpolation of actual values from runs at nearby compressions.

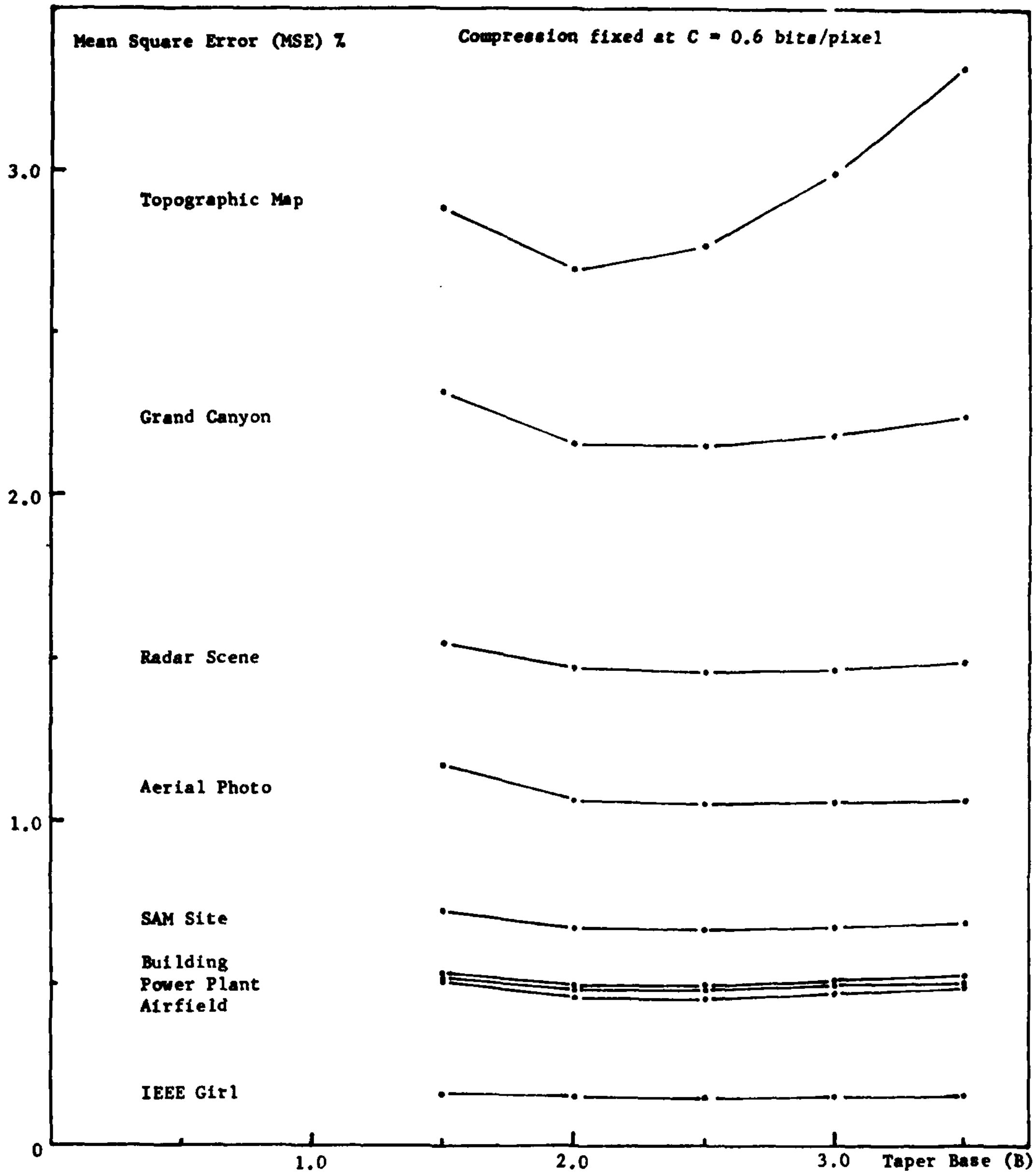


Figure 3-3. Mean Square Error vs Taper Base

- The mean-square-error (MSE) fidelity measure at fixed compression shows a very shallow minimum as a function of taper base near B=2.5. Anywhere within the range from B=2.0 to B=3.0 appears reasonable.
- Subjective evaluation of decompressed images at one fixed compression ($C \approx 0.6$ bits/pixel) suggests that visual fidelity is slightly better toward B=3.0 rather than at the taper ($B \approx 2.5$) corresponding to minimum MSE.

For the subjective evaluation, compressions as close as possible to C=0.6 bits/pixel where decompressed and photo written for each of the nine test frames at each of four tapers (B=2.0, 2.5, 3.0, and 3.5). Exact compression values cannot be reached, of course, because the coding is image adaptive based on contrast fidelity control. However, the mean compression over the thirty-six image-taper combinations was $\bar{C} = 0.601$ bits per pixel with a standard deviation of $\sigma_c = 0.016$ bits per pixel (2.6%). The worst case was only 5.6% away from the target compression. This imagery is displayed as Figure 3-4.

The final observation above is based on light table examination of the resultant positive transparencies of the decompressed images. As the taper is increased, the contrast scale (T_o) must also be raised to maintain fixed compression. Thus, the thresholds for very small elements are increased while those for the largest elements decrease. In essence, some small high contrast detail is sacrificed to improve the fidelity of mid-sized lower contrast features. Since the very smallest elements are relatively so expensive to preserve, deletion of a small amount of fine detail results in large gains in the mid-range. This appears to be the source of the visual preference for tapers slightly above those giving the optimum MSE.

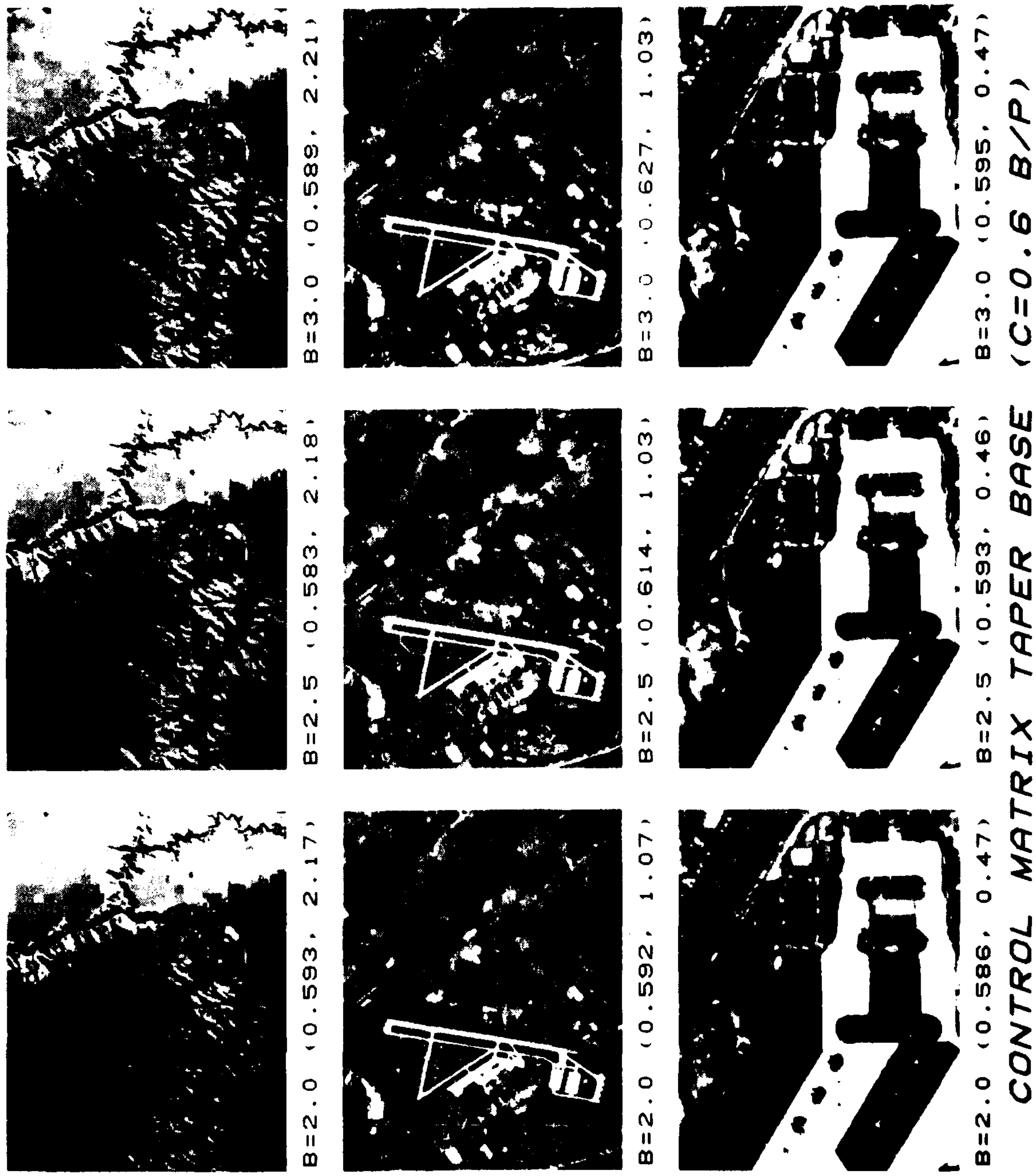


Figure 3-4 (a). Examples of Taper Base (B) Variation
(Compression in bits/pixel, MSE in %)

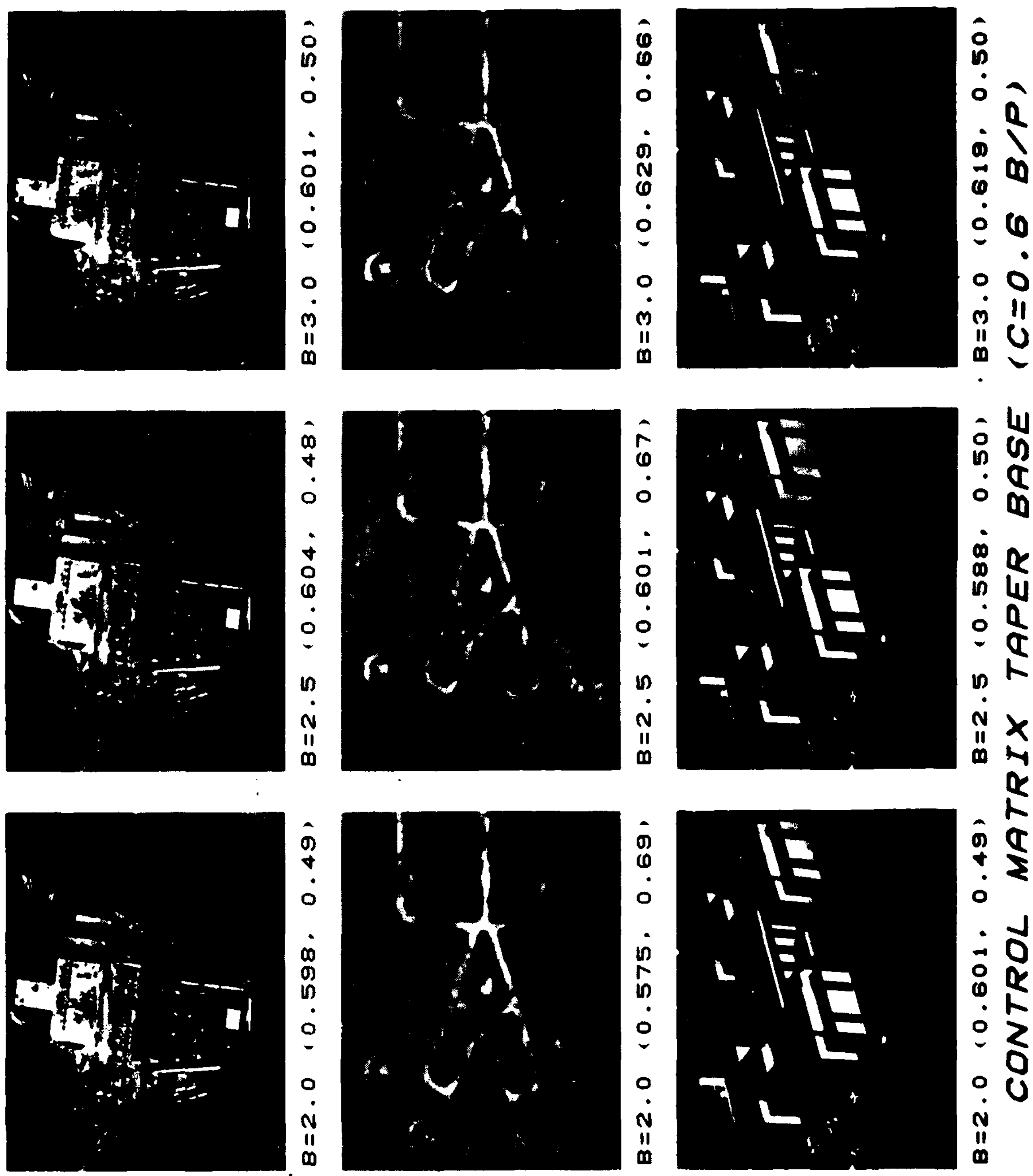


Figure 3-4 (b). Examples of Taper Base Variation (Continued)

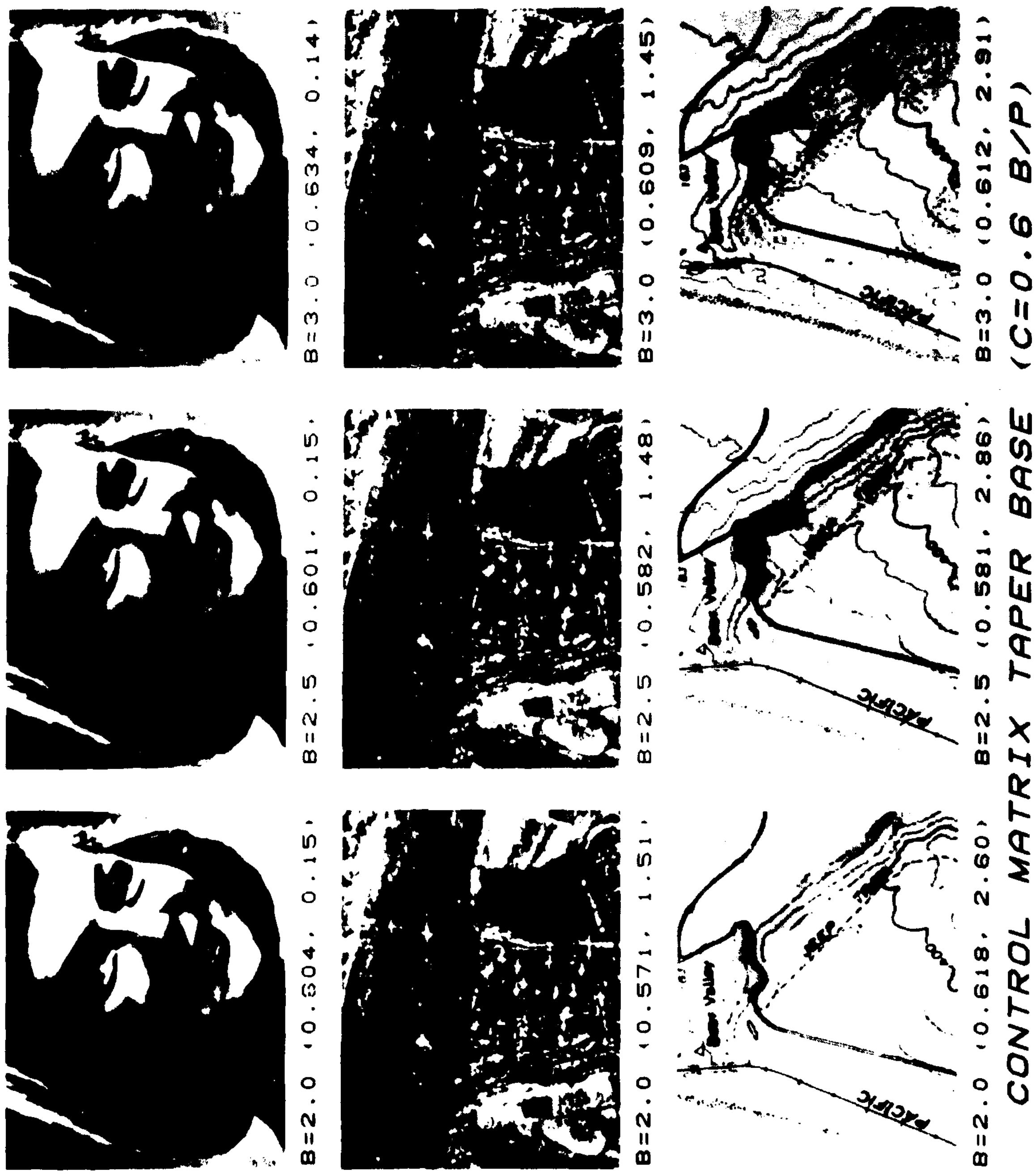


Figure 3-4 (c). Examples of Taper Base Variation (Continued)

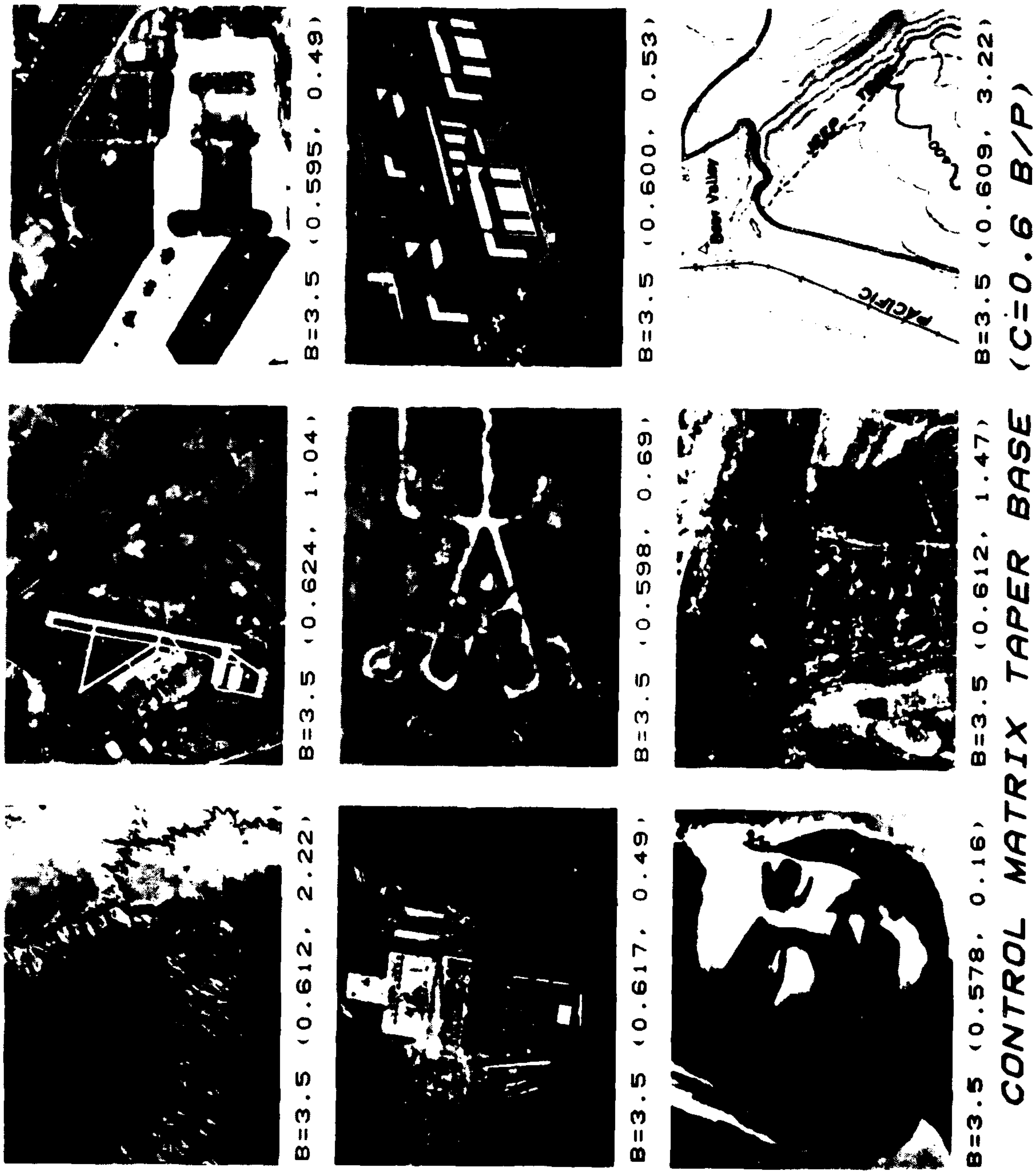


Figure 3-4 (d). Examples of Taper Base Variation (Concluded)

The following key conclusion is supported by these observations:

A UNIVERSAL MAPS EXPONENTIAL TAPER MAY BE CHOSEN.

A sample of the parametric change in taper base, B, for one of the test images is given in Figure 3-5. Note that the compression vs contrast scale relation changes systematically with taper but the function is not particularly smooth within a fixed taper. This will be discussed more extensively at the end of this section. All of the curves, however, are approximately log-log linear so no significant preferred value for this taper is implied.

3.2.2 Step Fraction and Outer/Middle Bias Trade-Offs

Four test frames — AERIAL PHOTO, AIRFIELD, IEEE GIRL and TOPOGRAPHIC MAP — were chosen for the F and Δ explorations. The taper base, B, was fixed at 3. Nine combinations of F and Δ were used:

$$F = 0.4, 0.5, 0.6;$$
$$\Delta = 0.0, 0.1, 0.2.$$

For each test image at each (F,Δ), the contrast scale, T_o, was searched to yield a compression as close to 0.6 bits per pixel as possible. The thirty-six samples finally chosen were then photowritten as positive transparencies for visual inspection. These images are presented as Figure 3-6.

The mean square error (MSE) versus compression curves show even less variation over this (F,Δ) range than the very weak variation observed as a function of B (at fixed F and Δ). Visual examination also reveals no marked (F,Δ) effects although each extreme value exhibits slightly discernible relative degradation on one or more of the test scenes. From these observations, the following conclusion is drawn:

UNIVERSAL VALUES OF STEP CONTRAST FRACTION, F=0.5, AND MIDDLE-OUTER STEP BIAS, Δ=0.1, ARE APPROPRIATE FOR MAPS CONTRAST CONTROL GENERATION.

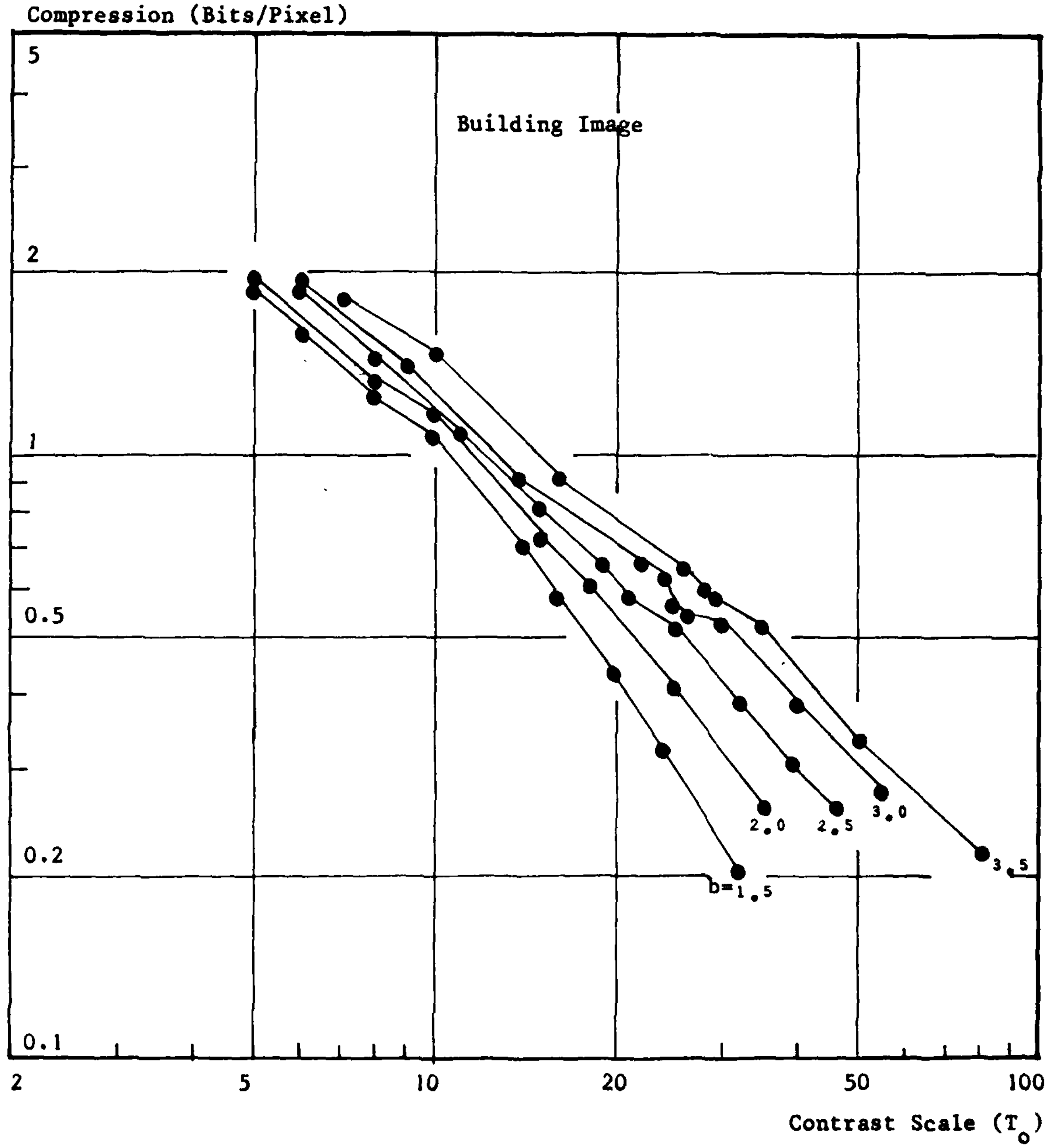


Figure 3-5. Compression vs Contrast Scale with Taper Base Parametric.

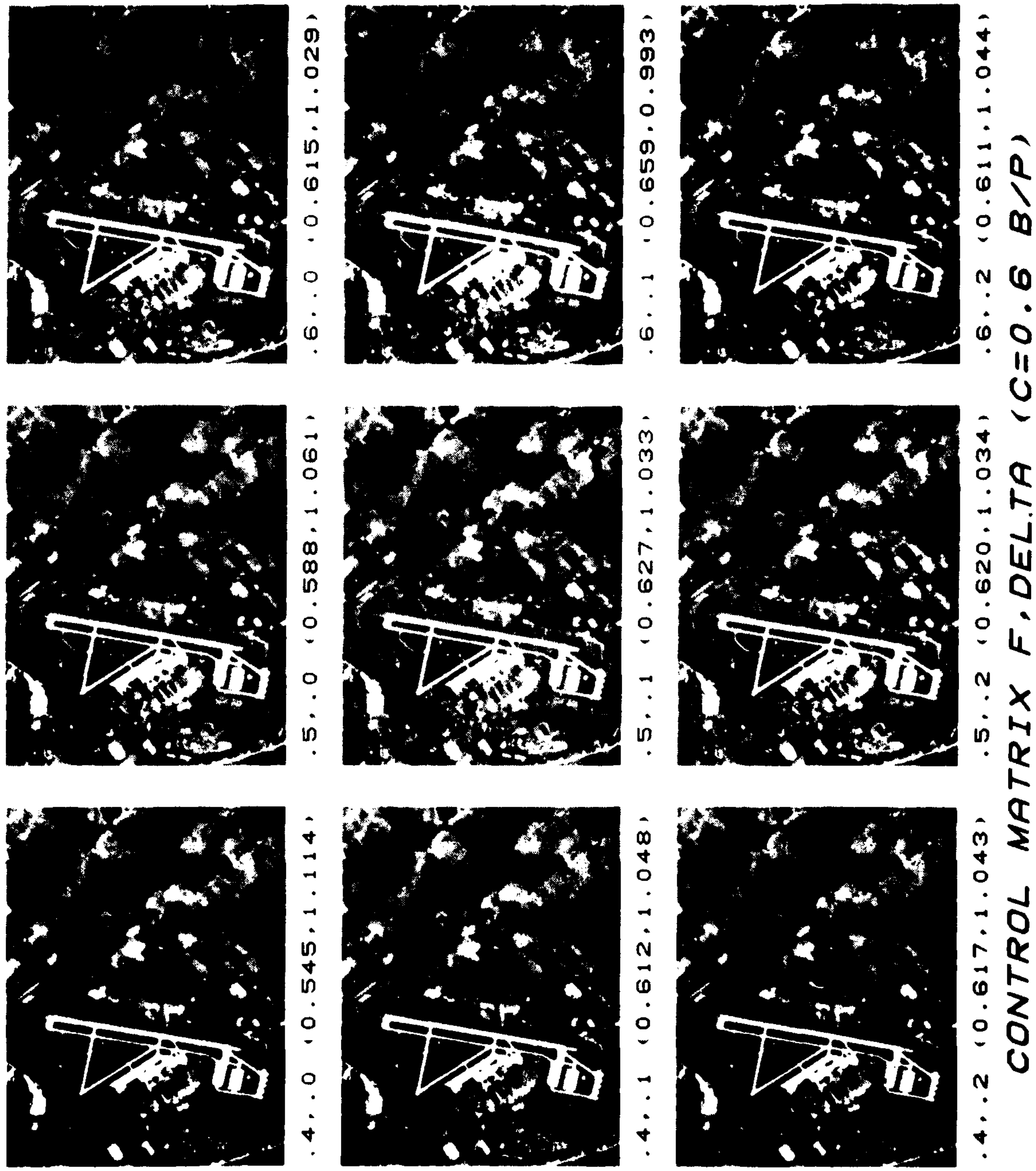


Figure 3-6 (a). Examples of Step Fraction (F), Outer/Middle Bias (Δ) Variation. Captions: F, Δ (Compression b/p, MSE %)

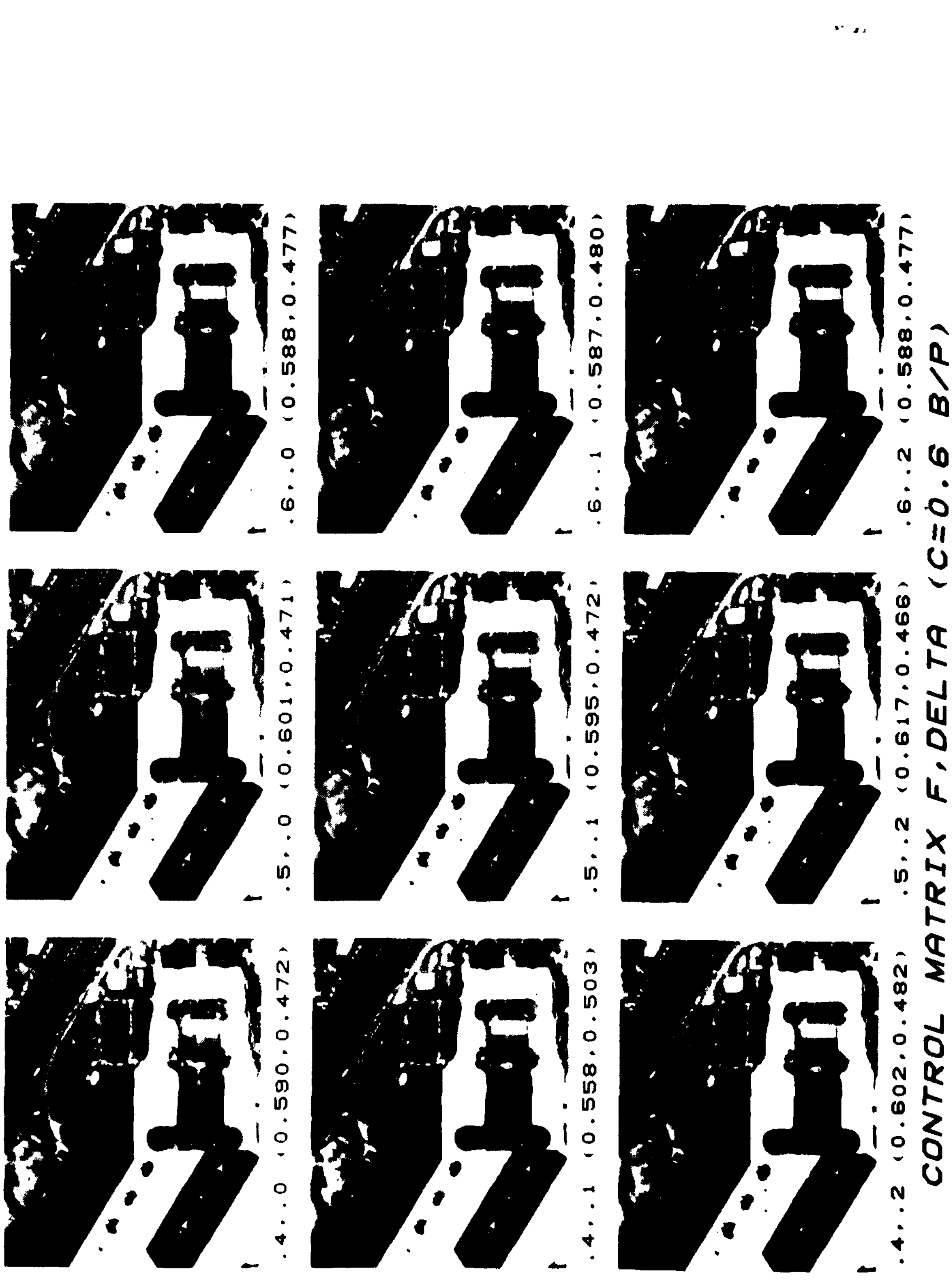


Figure 3-6 (b). Examples of Step Fraction, Outer/Middle Bias (Continued)

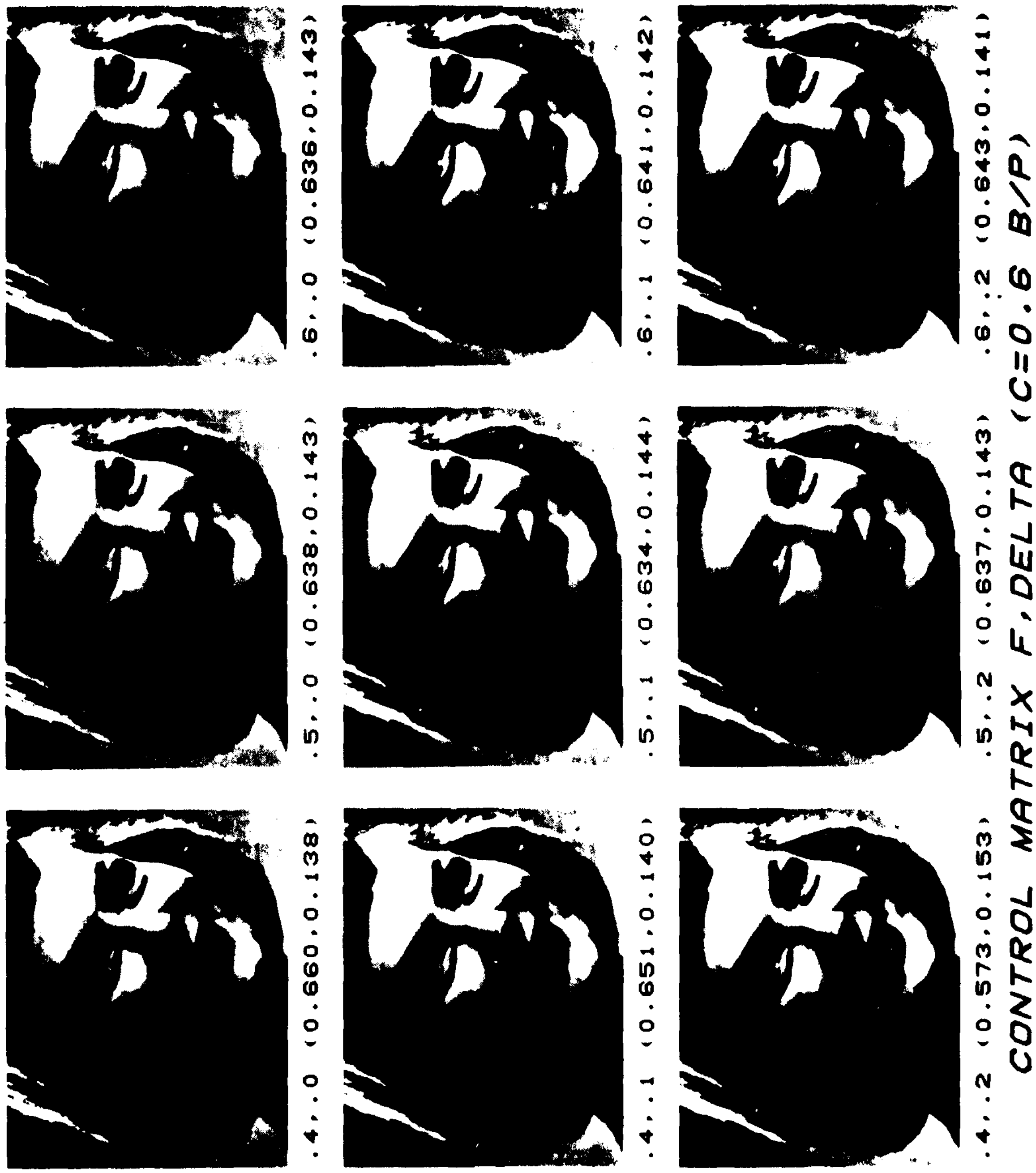


Figure 3-6 (c). Examples of Step Fraction, Outer/Middle Bias (Continued)

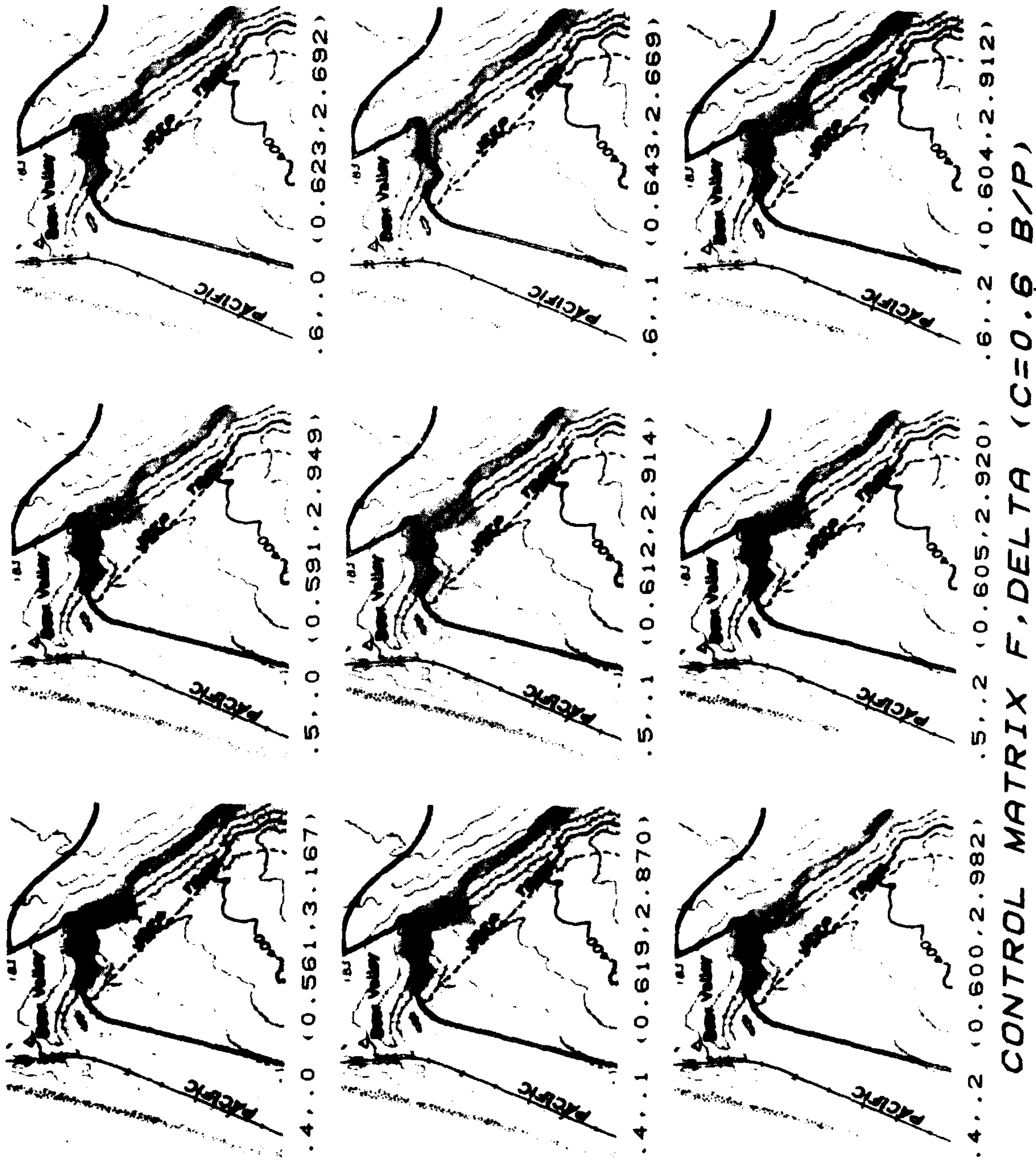


Figure 3-6 (d). Examples of Step Fraction, Outer/Middle Bias (Concluded)

Thus, only the contrast scale, T_o, remains imagery dependent — a significant reduction in control dimensionality from sixteen parameters to one!

3.2.3 Contrast Scale Dependence

The variation of the compression level with the contrast scale, T_o, is illustrated for fixed (B, F, Δ) in Figure 3-7. Here the test image scene is parametric. Note that the curves are all approximately linear and approximately parallel in this log-log space. As a consequence, the data base was analyzed to obtain linear least-squares fits of log C vs log T_o for each image at each taper with (F=0.5, Δ=0.1). The results are listed as Table 3-2 in terms of the slope.

Given any two (compression, contrast scale) pairs, the linear log-log fit corresponds to a relation

$$C' \approx \left(\frac{T_o'}{T_o}\right)^a C \qquad (3.2.3\text{-}1)$$

where a is the tabulated slope. Given a selection for the taper, relation 3.2.3-1 becomes a "rule of thumb" for extrapolating one (C, T_o) pair to another. The corresponding mean slope from Table 3-2 is a reasonable first estimate for the exponent, a. Finally, note that taper B=3 yields $a \approx -1$ or CT_o approximately constant.

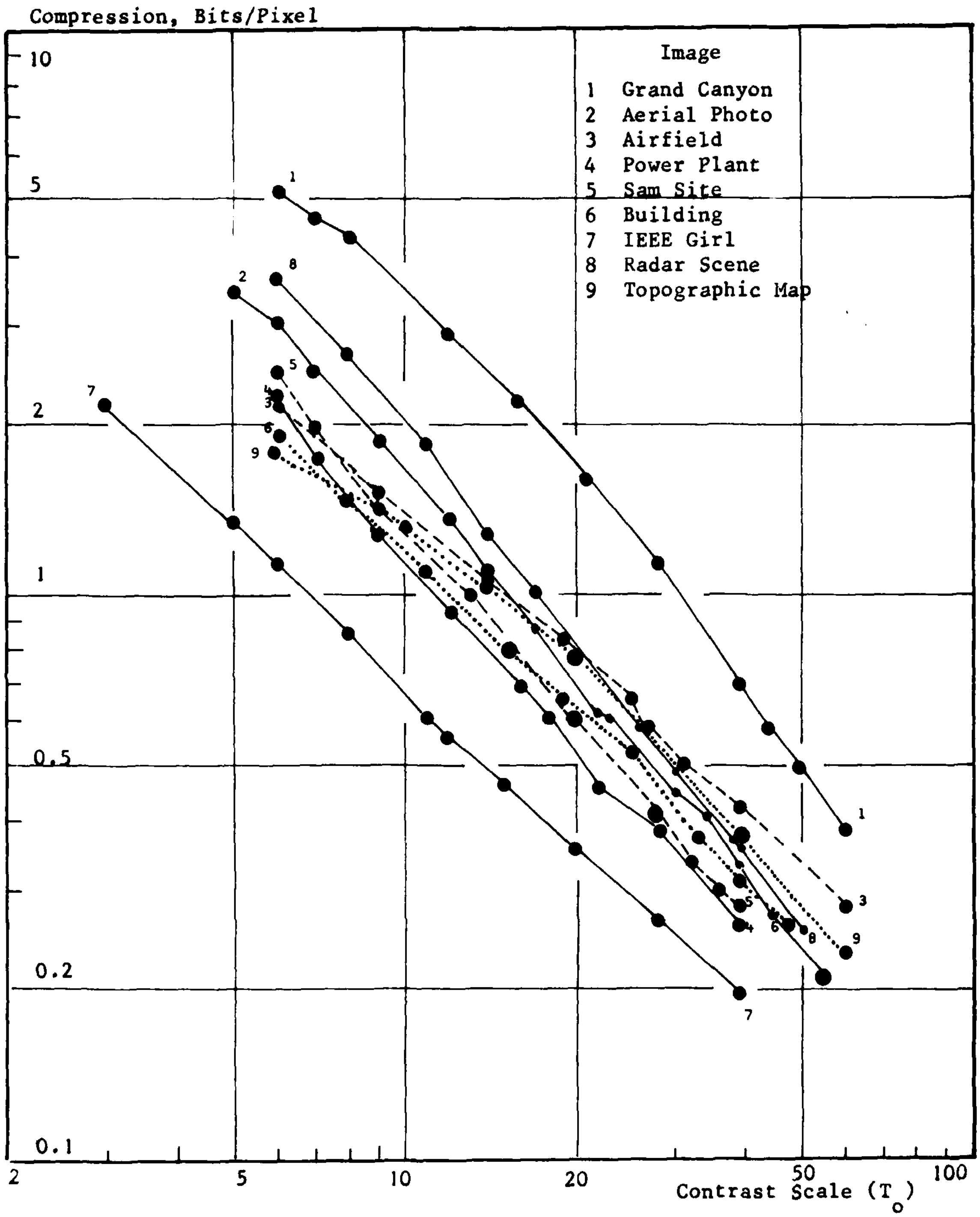


Figure 3-7. Compression vs Contrast Scale with Imagery Parametric, (B-2.5, F=0.5, Δ=0.1).

TABLE 3-2. LEAST SQUARES SLOPE OF LOG C (BITS/PIXEL) VS. LOG T_o WITH TAPER BASE PARAMETRIC

Image	Taper Base, B				
	1.5	2.0	2.5	3.0	3.5
Grand Canyon	-1.28	-1.50	-1.15	-1.20	-1.10
Aerial Photo	-1.65	-1.40	-1.18	-1.07	-0.95
Airfield	-1.15	-0.96	-0.88	-0.81	-0.77
Power Plant	-1.49	-1.26	-1.13	-1.04	-0.97
Sam Site	-1.75	-1.37	-1.15	-1.00	-0.94
Building	-1.17	-1.05	-0.97	-0.86	-0.84
IEEE Girl	-1.27	-1.04	-0.94	-0.89	-0.76
Radar Scene	-1.75	-1.56	-1.27	-1.10	-0.98
Topo Map	-1.06	-0.87	-0.89	-0.84	-0.80
Mean	-1.40	-1.22	-1.06	-0.98	-0.90
Sigma	0.27	0.25	0.14	0.13	0.11

SECTION FOUR
RESOLUTION CODE ERROR DETECTION AND RECOVERY

There are three levels of MAPS errors:

- intensity errors,
- intrasubframe position and resolution errors,
- intersubframe position errors.

These are ordered by increasing significance with the last class potentially catastrophic. However, the MAPS resolution codes retain some natural redundancy which implies an internal consistency which can be used for partial error checking. Here that redundancy will be organized to emphasize protection against long-range intersubframe error propagation.

The basic MAPS element (or "MAPSel") is localized as a compound word with N intensity bits and K resolution or level bits. Typically, N is six or eight and K is three for a 16x16 pixel subframe partition. However, only five of the eight codes available from these three bits are required — those corresponding to levels 0(1x1), 1(2x2), 2(4x4), 3(8x8), and 4(16x16). Codes for these levels need not be assigned to their natural binary representations. Moreover, when there are no errors within the subframe, the level for the final element in the subframe is implied by the number of original pixels remaining to be described by that element. Thus, the normal level code for the final MAPSel may be replaced by a "subframe termination" code. Note that this process does not use up any additional code values because the level four resolution code will be converted to the terminator — it is the last and only element in the subframe.

The recommended code assignments are L=0→0, L=1→1, L=2→2, L=3→4, and terminator (including L=4) → 7. This assignment is depicted geometrically in the code "cube" of Figure 4-1. There the motivation for the assignments can be seen more clearly:

- All codes above level zero are "Hamming distance 2" apart — at least two bits of three must be in error before any two of these codes are interchanged,
- All codes are "Hamming distance 2" <u>or</u> <u>more</u> away from the terminator — any single bit error in a terminator is immediately detected as an illegal code,
- Level zero is only 1 bit different from each of levels one, two, and three but level zero MAPSels must come in strings of four — this information can be invoked to detect errors in interchanges of level zero with other allowed codes.

Note that the termination code is thus "protected" at the one-bit error level, both in terms of addition of terminators by erroneous conversion from other legitimate codes and from undetected deletion.

The subframe integrity can be increased significantly by exploiting another global constraint on the MAPS level patterns. Each time a higher level MAPSel is formed it replaces four MAPSels at the immediately prior level. Hence, each composite element formation results in a net decrease of three in subframe MAPSel count. For a 16x16 subframe, the count ranges from 256 at full resolution down to one MAPSel at complete reduction. Only values of the form (3j+1), j=0, 1, ... 85, are allowed for the total subframe count. Expressed in a slightly different form, "the MAPSel count <u>between</u> terminators must be one of eighty-six multiples of three". This implies that a string of three terminators would all have to be lost (and the total element count of the three would have to fall below the bound) before such deletion would be undetected. Analogously, a total of three false terminators would have

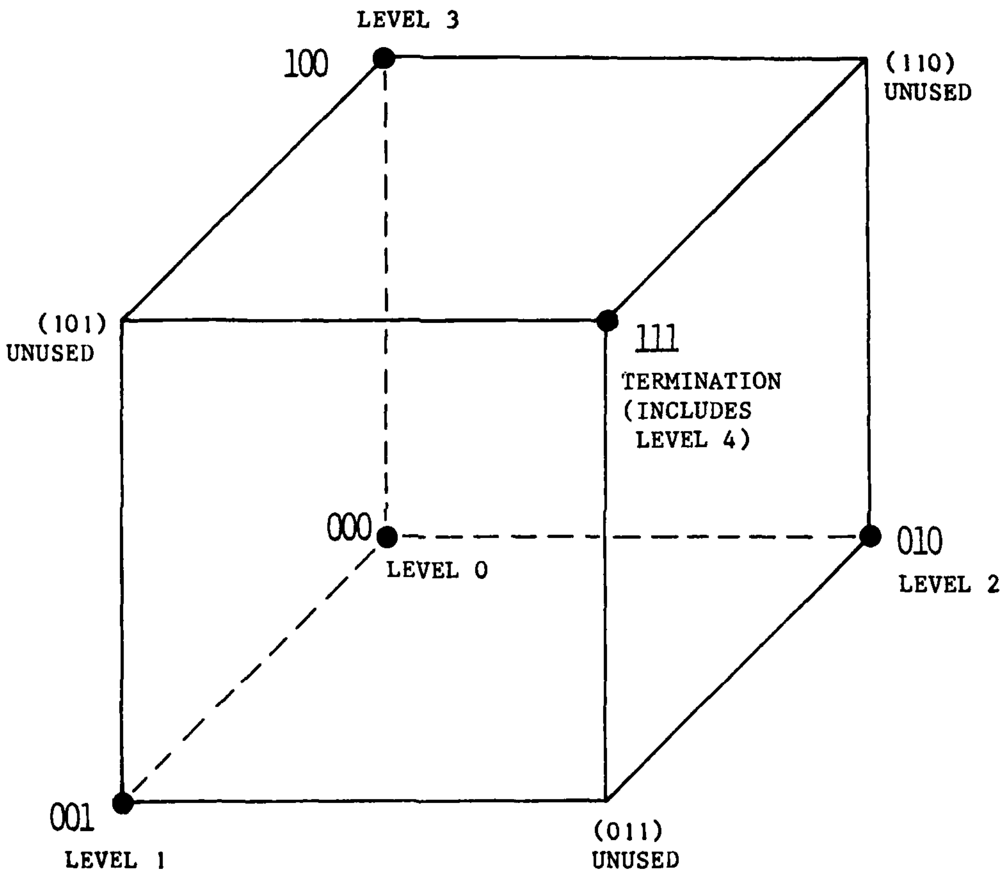


Figure 4-1. Resolution Code "Cube"

to be inserted between two legitimate terminator before such addition would be undetected. Moreover, each would have to be inserted in the stream at positions satisfying the "multiple of three" requirement — this has an a priori conditional probability of $(1/3)^3 = 0.037$ given the three false terminators in the interval. Note that a combination of the loss of a legitimate terminator and insertion of a false terminator (satisfying the "multiple of three") could also go undetected. In this case, the error would propagate across the first subframe boundary but not across the second.

Together, the intracode separation/detection and the intra/inter subframe "multiple of three" relation represent a capability similar to a horizontal and vertical parity system operating with emphasis on terminator integrity protection. Moreover, this is all achieved at no cost to the basic MAPS compression performance.

In addition to checking for simple illegal codes, the MAPS resolution level consistency requirements can be further exploited for local error detection. Given a legitimate position in the subframe, only certain resolution levels (and codes) are allowed for the next MAPSel. The allowed levels are summarized in Figure 4-2(a) and the corresponding codes (with the terminator included) in Figure 4-2(b). An erroneous code may violate the constraint and be detected immediately or it may throw off the phase of the inferred position with respect to the true position and result in a violation later. When an illegal code is encountered, the various allowed codes may be tried in the order of their "distance" from the observed value. Together, the code assignment and constraint map provide a significant local detection and recovery potential.

All techniques above take advantage of constraints and redundancy already present in the basic MAPS code. If the environment is very noisy, addition of further redundancy for error detection and correction is required. Retention of the predictability of position of the resolution data in the

0	1	0	2	0	1	0	3	0	1	0	2	0	1	0	4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	2	0	1	0	2	0	1	0	2	0	1	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	2	0	1	0	3	0	1	0	1	0	1	0	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	2	0	1	0	2	0	1	0	2	0	1	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a) Maximum <u>next</u> resolution level vs position in subframe

							(010)				(001)		(000)	(***)	(100)
000	001	000	010	000	001	000	111	000	001	000	111	000	111	111	111
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	001	000	001	000	001	000	001	000	001	000	001	000	001	000	001
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	001	000	010	000	001	000	010	000	001	000	010	000	001	000	010
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	001	000	001	000	001	000	001	000	001	000	001	000	001	000	001
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	001	000	010	000	001	000	100	000	001	000	010	000	001	000	100
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	001	000	001	000	001	000	001	000	001	000	001	000	001	000	001
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	001	000	010	000	001	000	010	000	001	000	010	000	001	000	010
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	001	000	001	000	001	000	001	000	001	000	001	000	001	000	001
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000

Figure 4-2 (b). Maximum <u>next</u> resolution code vs position in subframe

bit stream is still required which means that a uniform element format is mandated. However, since additional redundancy is to be included, that associated with the unused level codes might first be eliminated. Note that each code requires $\log_2 5=2.32193$ bits (rather than three) and a combination of resolution codes for three elements can thus be described in 6.9658 bits ($5^3=125 < 128=2^7$). Thus a seven-bit code can be used as the basic resolution element (rather than three three-bit codes). The coding then gives this seven bit quantity augmented by some error correction bits plus three N bit intensity values as the basic MAPS data packet. Alternatively, several seven bit representations of three resolution codes may be strung together and protected by a more elaborate code (but with fewer bits). Note that multiples of seven bits are required until 90 cells have been joined wherein 209 bits (rather than 210) would suffice.

One very simple alternative with the three cell/seven bit combination involves a single parity bit with each triple. This makes an eight-bit resolution code format or a thirty-two bit intensity/resolution code packet if there are N=8 intensity bits per MAPSel. This is a particularly attractive size for many current digital systems. Note that the compression ratio is actually increased a little in this case (by 33/32). The individual-element code separation has been lost here but an explicit local parity check has been established on three elements. Moreover, use of the "multiple of three" termination interspace relation and the next-resolution-code constraint map are preserved. Note finally that for a burst error environment, the resolution code bits may be interleaved with the intensity bits to reduce their adjacency (at the cost of packing and unpacking manipulations). For the three-MAPSel/thirty-two bit packets described above, resolution code bits may be placed in every fourth location.

SECTION FIVE

ON-LINE MAPS DEMONSTRATION/ INTERACTIVE MACRO-FIDELITY CONTROL

The MAPS compression and block decompression techniques were implemented on a high performance image processing system. A user can load any image into the system, from tape, film, or disk. Compression parameters can be input to the system, and in less than 2 seconds the system can compress and decompress the image. The display facility allows the user to view both the original and the processed image in a variety of display modes, including zoom, split-screen, and with color graphics. Macrofidelity control is implemented using a trackball to create arbitrary areas for fidelity control. Up to four different T_o parameters may be specified to allow four levels of fidelity in the processed image. These capabilities allow the user to see the efficiency and adaptability of the MAPS technique. The rapid processing allows the user to experiment quickly with different compression parameters and see the trade-offs between compression and fidelity.

5.1 IMAGE PROCESSING SYSTEM DESCRIPTION

The image processing system consists of a CDC 1700 system linked to a four-flexible-processor (FP) array. The 1700 provides high-level language programming and operating system functions to host the FP's. The FP array provides for general purpose display functions such as trackball operation and generation of scan conversion parameters. Additionally, the FP provides high speed computation for real time image processing such as MAPS. A block diagram of the system is shown in Figure 5-1.

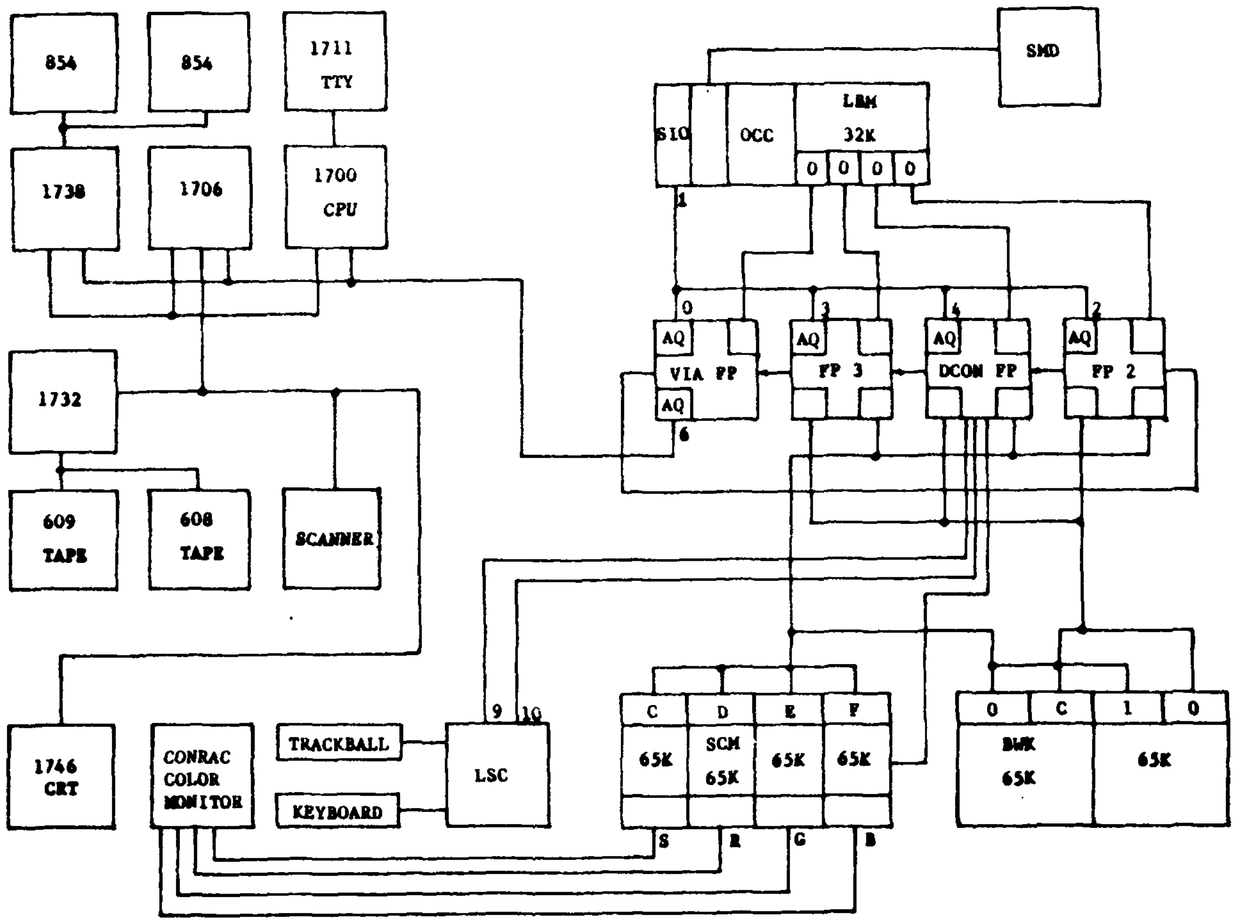


Figure 5-1. Image Processing System Block Diagram

The 1700 system consists basically of a 1700 central processor, two tape drives, two disk drives, a C.R.T., and an optronics scanner, with their associated controllers. This system provides utilities to aid in FP microcode checkout and to perform I/O for the FP array. Utilities exist to transfer imagery from tape or film, via the scanner, to the FP array and to load microprograms into the FP's. These utilities can be called directly by the operator or they can be called by a FORTRAN driver. Additionally the 1700 is used to create and maintain the library of microcode which is used in the FP array.

The heart of the image processing system is the Scan Converter Memory (SCM) and the FP array. The SCM can hold more than a half million bytes of image data and drives a 640 x 512 color monitor; additionally the color conversion and raster map can be changed up to 30 times per second. The display controller (DCON) routine provides user interaction with SCM by user microcode overlays, by low-speed channel (LSC) trackball and keyboard input, or by 1700 input. Also linked to the FP array is a storage module drive (SMD) which can hold more than 60 million bytes of image data. This allows the user to roll images in and out of SCM in about 2 seconds. Using DCON, imagery can be displayed in a variety of modes including zoom, split-screen, and color. In addition, graphics can be overlayed on imagery. A more complete description of the FP and SCM is given in Appendix A.

Microcode was developed to perform both the on-line demo and macro-fidelity control tasks. The on-line demo is incorporated in the macro-fidelity control program. This allows the user to input up to four T_o parameters and arbitrarily assign one of the corresponding contrast control matrices (CCM's) to each block in the image, then compress and decompress the image and display the compression ratio. The original and processed images are stored separately in SCM so that both can be viewed. Using the display facilities of DCON, the images can be independently zoomed and viewed using split-screen; and displayed using a variety of color look-up tables. The

SCM and microcode can be loaded using INDIAN2 utilities. INDIAN2 is the 1700 display control software which complements DCON in the FP array.

5.2 MAPS MICROCODE SOFTWARE

The code which performs the MAPS compression and decompression consists of 5 parts: an executive (MAPX), a read routine (REDB), a compression routine (COMB), a decompression routine (DECB), and a write routine (WRTB). MAPX requires a 512 pixel x 480 line x 8 bit image stored in SCM as input. The lower 5 bits of the pixel are treated as the image and the upper 3 bits are ignored. The output of MAPX is a 512 pixel x 480 line x 8 bit image in SCM where the lower 5 bits of a pixel are the image grey scale and the upper 3 bits are the resolution code of the parent MAPS pixel. The other four routines operate at the block (16 pixels x 16 lines) level. Input to, and output from, each routine is a block. The flow of data is summarized in Figure 5-2.

The operation proceeds at about 35.3 instructions/pixel overall. An enhanced version could run at about 25.6 instructions/pixel. All the enhancements would require more LF storage and instruction memory.

5.2.1 Flexible Processor I/O (REDB, WRTB)

The (I/O) routines REDB and WRTB basically perform two functions; transferring data to and from SCM and converting raster format to and from MAPS order. Additionally REDB clears the top three bits of the pixels and communicates macrofidelity control information to MAPX. This allows these three bits to be used for graphics. Both routines were written to operate near the bandwidth of the SCM; approximately 2 instructions/pixel. WRTB runs at about 3.0 instructions/pixel and REDB runs at about 3.7 instructions/pixel. REDB is slower due to the clearing of the top bits of the pixels.

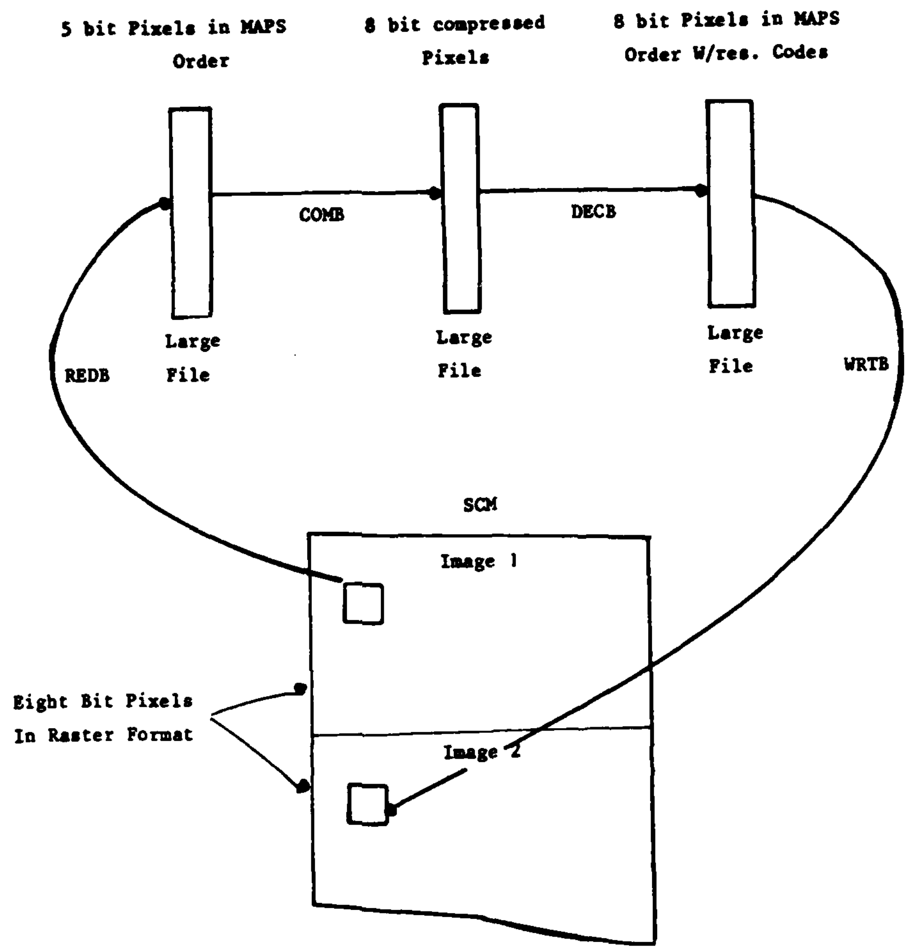


Figure 5-2. MAPS/FP Data Flow Chart

5.2.2 MAPS Compression (COMB)

COMB is the heart of the MAPS/FP implementation and uses a different algorithm to perform compression than its FORTRAN predecessors. The input to COMB is in MAPS order and the output of COMB is not bit-stream. The purpose of creating this new algorithm was to find a fast implementation for the FP. This meant keeping the number of conditional jumps to a minimum and using sequential rather than random access to pixels in large file (LF). COMB achieves this by using an iterative rather than recursive algorithm and performing the pixel sort with a look-up table. The result is that COMB operates at about 25.4 instructions/input pixel, and further enhancements could lower the rate to about 17 instructions/input pixel.

In order to develop a fast compression routine, an iterative compression algorithm using a new sort was developed. The pixel sort is implemented using a look-up table rather than the tree sort used before. A tree sort requires 4.67 differences on the average; however, in some cases, an additional difference is required for contrast control comparisons. Actually 5.33 differences are required on the average. The new sort takes all six differences of four, taken two at a time, and forms a 6 bit address from the sign bits. This is the address in a look-up table of the extreme, middle, high, and low step differences required for contrast control comparison. This method requires no conditional jumps where a tree sort requires 4.67 on the average. In order to make the algorithm iterative, COMB makes 4 passes over the data. Previously, pixels were stacked pending further compression; COMB does all compression at one level before compressing at the next level, thereby eliminating the stack. The COMB algorithm is shown in Figure 5-3. These changes were made to eliminate all random access to pixels in LF.

5.2.3 MAPS Block Decompression (DECB)

DECB performs block decompression on compressed pixels. Its operation is a straightforward extension of the FORTRAN versions. Additionally DECB

//COMB//

```
        P←1;    //PASS NUMBER//
        n←0;
        A←0;   //LF address; (LF)_A  ≡ contents of location A in LF//
        a←4^(p-1);
START       n←n+1;
            P1←(LF)_A;  A←A+a;  //P_i, i=1-4 are the 4 pixels to compress//
            P2←(LF)_A;  A←A+a;
            P3←(LF)_A;  A←A+a;
            P4←(LF)_A;
            if [all 4 pixels P1-P4 are not level p-1]  then go to LOOP1;
            sort pixel and locate E, M, H, 6 steps;
            if [Extreme threshold violated]  then go to LOOP1;
            if [Middle threshold violated ]  then go to LOOP1;
            if [Low threshold violated    ]  then go to LOOP1;
            if [High threshold violated   ]  then go to LOOP1;
            (LF)_A ←  P1 + P2 + P3 + P4 + 2) / 4 + $20;
            // $ 20 is the hexadecimal value used to increment //
            // pixels from level p-1 to level p               //
LOOP1       A←A+a;
            if [n < 4 ^(4 -p)] then go to START;  // pass not complete//
        if [p = 4]  then EXIT;
        p ← p + 1;          // start next pass //
        n ← 0    ;
        A ← 0
        a ← 4 ^(p-1);
        go to START;
```

Figure 5-3. MAPS/FP Compression Routine //COMB//

accumulates a compressed pixel count for compression ratio calculation. DECB runs at about 3.2 instructions/output pixel and could be enhanced to run at about 2.1 instructions/output pixel.

5.2.4 Macro-Fidelity Control

The code to perform macrofidelity control is largely involved in calling routines in DCON which perform display and arithmetic primitives. These routines perform character generation, line drawing, clearing character lines, turning the cursor on and off, clearing specific bits in each pixel, and multiplication and division. Additional primitive routines were developed for converting fixed point binary numbers to ASCII strings and vice versa. The mechanism used to communicate T_o values to the compression routine for macrofidelity control uses the top 2 bits of the pixel in the original image. These bits perform two functions: transmitting T_o values, and designating the color to appear in the outline of each block. Drawing the colored outline around a block during area definition actually sets the top 2 bits of those pixels making-up the colored lines. The REDB routine transmits these 2 bits to COMB.

Special coding considerations are elaborated in Appendix B and comment-annotated microcode is given in Appendix C.

5.3 USER OPERATIONS AND OPTIONS

A feeling for the general image manipulation and display capabilities available to the user can be obtained from a review of Figure 5-4. The controls and commands described therein can be used in concert with the more specific MAPS processes called from the display console. The MAPS operation consists basically of two parts: initialization and interaction.

TABLE 0 - imagery and graphics for IMAGE 1
TABLE 1 - imagery only for IMAGE 1 and 2
TABLE 2 - imagery with level 0 pixels in green for IMAGE 2
TABLE 3 - imagery with level 1 pixels in red for IMAGE 2
TABLE 4 - imagery with level 2 pixels in magenta for IMAGE 2
TABLE 5 - imagery with level 3 pixels in cyan for IMAGE 2
TABLE 6 - imagery with level 4 pixels in yellow for IMAGE 2
TABLE 7 - level codes in color
Blink vectors for pertinent images:
IMAGE 1 - 0, 1, 0, 1, 0, 1, 0, 1
IMAGE 2 - 1, 2, 3, 4, 5, 6, 7, 1

DEVICES: A device is a LSC input device. Devices can be linked to SCREENS, then an input from that device will cause that screen to be displayed.
DEVICE 1 - LSC keyboard
DEVICE 2 - trackball (one trackball may be eliminated)
DEVICE 3 - trackball

The pertinent commands which DCON accepts are listed below. If the keys are pressed in the order shown (from left to right), the system will perform the adjacent function.

Keyboard Sequence		Function
Screen X Image Y		links screen X to image Y
Device X Screen Y · Z		links device X to screens Y and Z (devices can be linked to 1 to 3 screens)
Cursor Mode V		cause the screens linked to the keyboard to have cursor mode V. The cursor modes possible (where V = CROSS, V = SCREEN, etc.), are
	CROSS:	trackball and arrows will move the cursor about the image
	SCREEN:	trackball and arrows will pan the screen across the image (this is usually used with the image zoomed)

Figure 5-4. User Commands for Image Display Utilities

Keyboard Sequence		Function
	WRITE DOTS:	this is the same as the cross mode except that in this case a trail of dots will be left by the cursor
	ERASE DOTS:	this is the same as the cross mode except that now the cursor can erase dots
	LINE:	trackball and arrows will move one end of a line extending the position of the last cursor mode update to the current cursor position.
	ZOOM:	trackball and arrows will zoom the image
	X:	limits cursor motion to the X direction
	Y:	limits cursor motion to the Y direction
	XY:	restores cursor motion to both the X and Y directions
	TILT:	trackball and arrows tilt the image
Zoom Mode, X		causes images linked to screens linked to the keyboard to be zoomed to the specified level and displayed
Output Mode, X		causes images linked to screens linked to the keyboard to be displayed using the look-up table indicated by position X in their blink vector.
Blink Step		cause images linked to screens linked to the keyboard to be displayed using the next look-up table indicated by their blink vector
Auto Blink		causes images linked to screens linked to the keyboard to blink step automatically at the preset time intervals.
Clear		Clears the input line.
Back Space		backspaces the input line one character.
Skip		skips forward one character on the input line.

Figure 5-4. User Commands for Image Display Utilities (continued)

5.3.1 Initialization

The initialization is carried out through user interaction with the 1700 on the operator's CRT and keyboard.

- RUN PREMAPS under the INDIAN2 subsystem of MSOS; this loads the appropriate look-up tables and image formats.
- LOAD IMAGEPY using INDIAN2 utilities. (for tape: IMDSKA3; for SMD: DEMO, DISPLA; for film: SCN, SCNTFR)
- RUN MAPS under INDIAN2; this loads the MAPS microcode.

5.3.2 Interaction

The remaining interaction takes place directly with the FP array on the LSC keyboard and trackball. The interaction proceeds in two phases: T_o selection and area definition. Information and prompts as well as user inputs are displayed in color on the original image throughout both phases. (Note: DCON's interpretation of keyboard inputs is sometimes dependent on previous inputs; to negate this possibility, press the RESET key before making inputs.) A table of T_o values is displayed in the upper left hand corner of the original image.

5.3.2.1 Contrast Scale Selection

- T_o SELECTION is indicated by the message INPUT T_o VALUE displayed in the lower left corner of the original image.
  - A pointer, ">", is displayed to the left of the table entry to be changed by the current input. Initially the pointer is displayed next to the T_o (Ø) entry. After each input, the pointer advances to the next entry until all four entries have been treated; then operation proceeds to the area definition phase.

- To change the indicated T_o value: input a two digit integer (inputs must be two digits: Ø1=1 etc.), then SEND. The new T_o value will appear in the table.
- To by-pass the indicated T_o value: just SEND. The pointer will advance without changing any table entries.

5.3.2.2 Macro-Fidelity Area Definition

- AREA DEFINITION is indicated by the message HOLD, MARK, SEND, GO in the lower left corner of the original image. The cursor should be turned on during the area definition phase. There are four meaningful keyboard inputs during area definition: M, H, SEND, G.
    - A pointer, ">", is displayed next to the T_o entry to be applied to area definition.
    - M (mark) followed by SEND will cause the FP to start marking blocks; whatever block the cursor enters will be given a color outline, the same color as the currently indicated T_o entry. This specifies that this block be compressed using the currently indicated T_o value (the default values for B, F, and Δ are 3., .5, .1 respectively). Initially no blocks are marked; this situation means that T_o (Ø) will be used for all blocks. (Although the T_o (Ø) entry is white, the blocks are not outlined. Hence marking with T_o (Ø), indicated by the pointer, will actually erase any marked blocks encountered).
    - H (hold) followed by SEND will cause the FP to stop marking blocks but will not advance the pointer. This allows the user to mark a disconnected area with the same color.
    - SEND alone will cause the pointer to advance to the next T_o entry. The table is essentially circular during area definition; that is after T_o (3) the pointer advances to T_o (Ø). Additionally, SEND alone, will cause the FP to hold if it is presently marking.

- G (go) followed by SEND will cause the FP to stop area definition and compress and decompress the image. Upon completion, the compression ratio is displayed to the right of the T_o (Ø) table entry. (The compression ratio displayed is for compression of 5 bit pixels to 8 bit pixels, the compression ratio would be about 7 percent larger for 6 bit to 9 bit compression).
- At this point the message CLEAR AREAS? will appear in the left lower corner of the original image.
  - Y (yes) followed by SEND will clear the defined areas and return to T_o selection phase.
  - N (no) followed by SEND will return to T_o selection with the defined areas intact.
  - SEND alone will stop the program and return the background schedule to DCON's control. This is necessary to load imagery.

5.4 DEMONSTRATION EXAMPLES

Figures 5-5, 5-6, and 5-7 show examples of images which have been compressed with the MAPS algorithm. The figures show the original imagery and imagery processed with and without macrofidelity control at a variety of compression ratios. A 'map' of compression is created with color graphics displaying the resolution codes of the output pixels. This technique demonstrates the adaptability of the MAPS algorithm and provides insight into its current and future applications. The interpretation of the colors is given in Table 5-1.

5.4.1 Graphics at Uniform Fidelity

Figure 5-5 shows the power plant original with and without graphics and a version at 5.2 to 1 compression displayed with 7 different look-up tables. No macrofidelity was used on this compression, so only the T_o (Ø) parameter is used. Frames C and A show the original image with and without

TABLE 5-1. COLOR CODES FOR RESOLUTION CODE DISPLAY

Color	Meaning
Green	level 0 pixels (1 pixel x 1 pixel , .625 compression ratio)
Red	level 1 pixels (2 pixels x 2 pixels, 2.5 compression ratio)
Magenta	level 2 pixels (4 pixels x 4 pixels, 10. compression ratio)
Cyan	level 3 pixels (8 pixels x 8 pixels, 40. compression ratio)
Yellow	level 4 pixels (16 pixels x 16 pixels, 160. compression ratio)

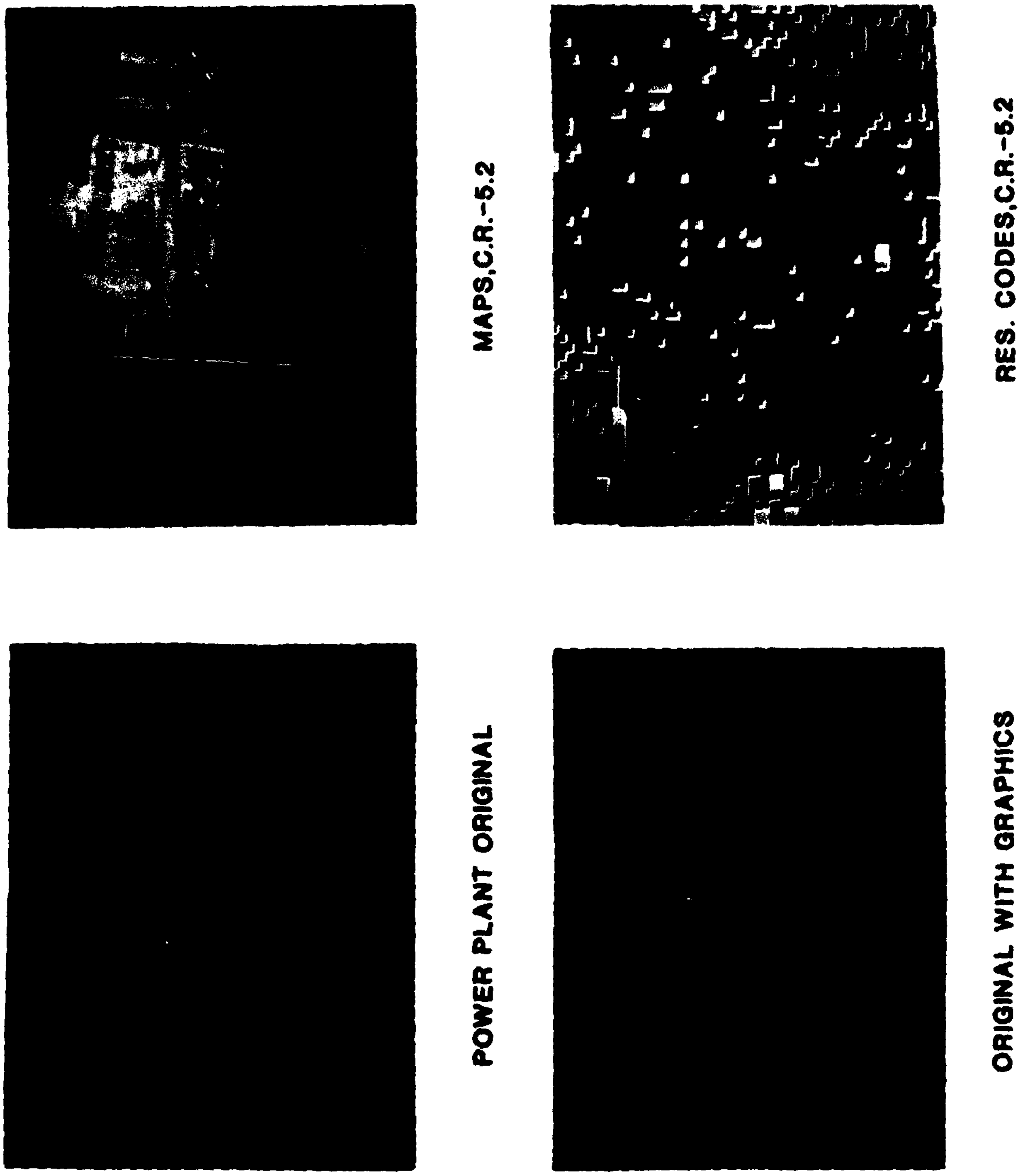
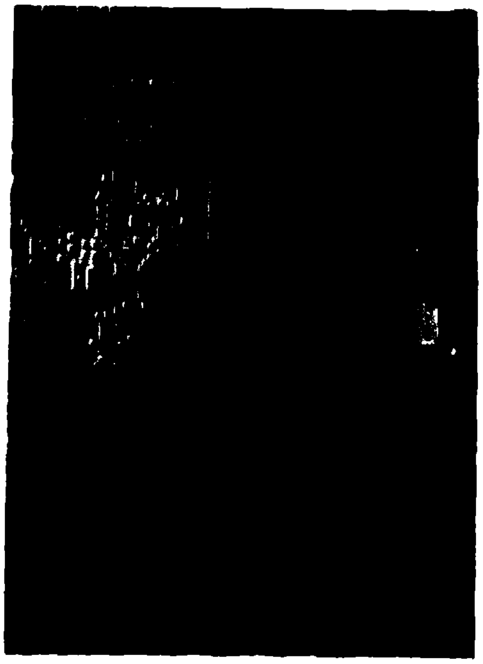
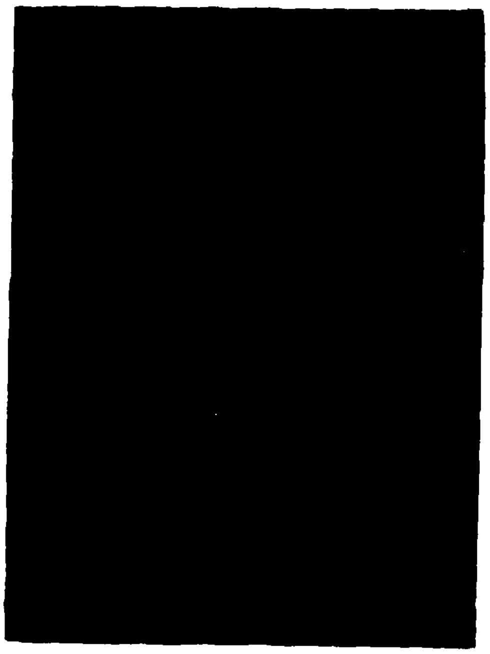


Figure 5-5. $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ Uniform MAPS Fidelity, Graphics Examples



LEVEL 1 PIXELS

LEVEL 3 PIXELS



LEVEL O PIXELS

LEVEL 2 PIXELS

Figure 5-5. $\begin{pmatrix} E & F \\ G & H \end{pmatrix}$ Graphics Examples (continued)

color graphics; frame C shows the display as the system waits for a new T_o (Ø) value to be input during the T_o selection phase. Frames B and D through H show the compressed image which has been converted using seven different look-up tables.

5.4.2 Compression Ratio Variation

Figure 5-6 shows the BUILDING scene at three different compression levels, without the use of macrofidelity control. Each compression level is represented by two frames showing the output imagery and the resolution codes. The resolution codes in these images indicate the potential for region growing and edge detection using MAPS. Frames B and F represent a compression ratio of 1.97 and $T_o = 4$, C and G represent a 5.2 to 1 compression with $T_o = 12$ and, D and H show 10.4 to 1 compression using $T_o = 25$.

5.4.3 Macro-Fidelity Variation with Split Screen, Zoom

Figure 5-7 shows the AERIAL PHOTO and contrasts compression with and without macrofidelity control. Frames B and F show the image compressed 6.8 to 1 with no macrofidelity controls. Frames D and H show the image compressed 6.8 to 1 with macrofidelity controls to preserve the area around the terminal. The degree of preservation is vividly displayed in the resolution codes in frame H. The macrofidelity controls are shown in color in frame C. There are four levels of fidelity achieved by using four T_o values: 28 (no color), 16 (yellow boxes), 10 (cyan boxes), 4 (magenta boxes). Frames E show a split screen with the original image on top at a zoom level of 3 and the image in frame D at zoom level 3 on the bottom. Frame G shows the image in frame D on top unzoomed and the image in frame H on the bottom unzoomed.

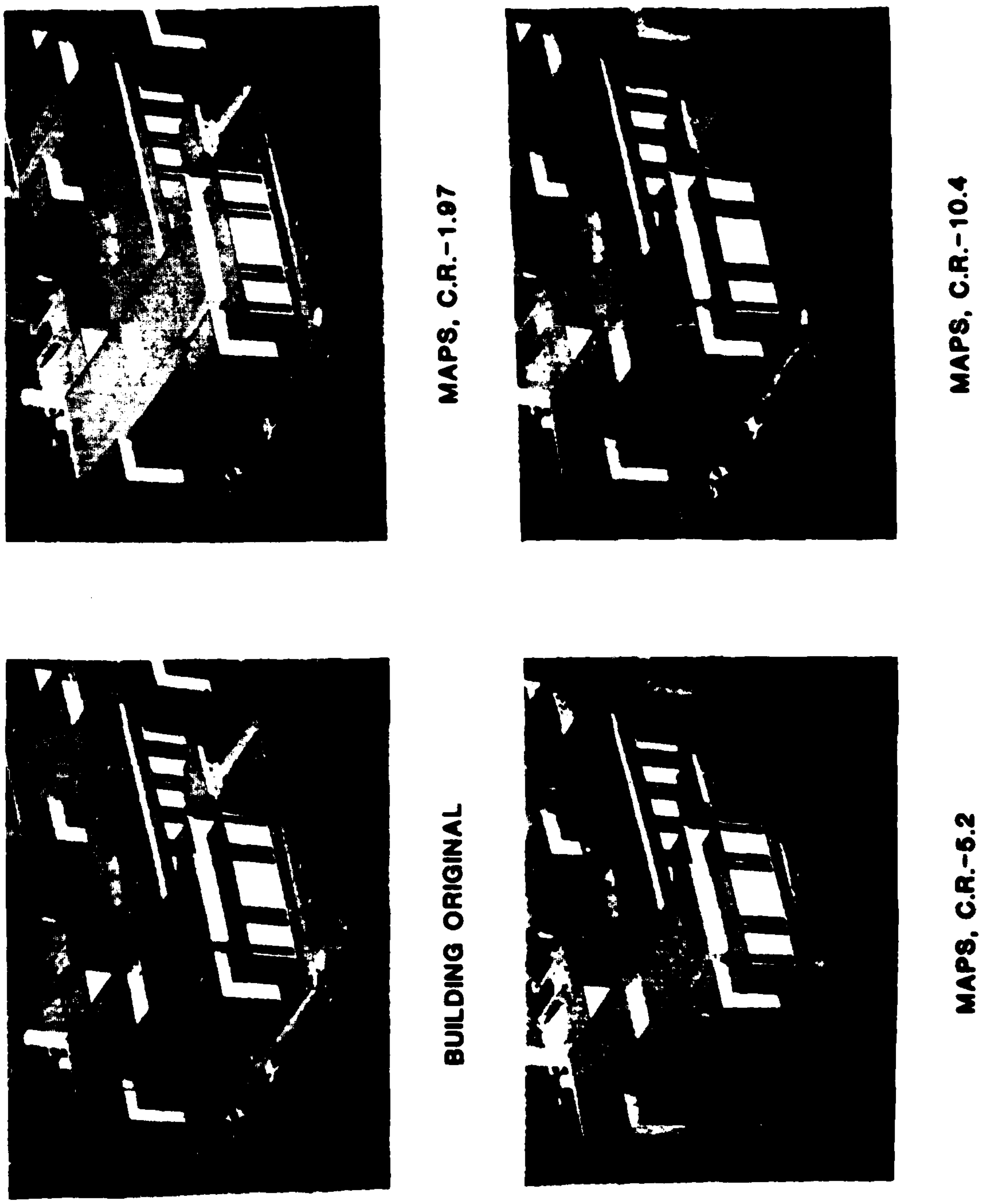


Figure 5-6 $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ Varying Compression Ratios, Tonal and Resolution Code Image Examples

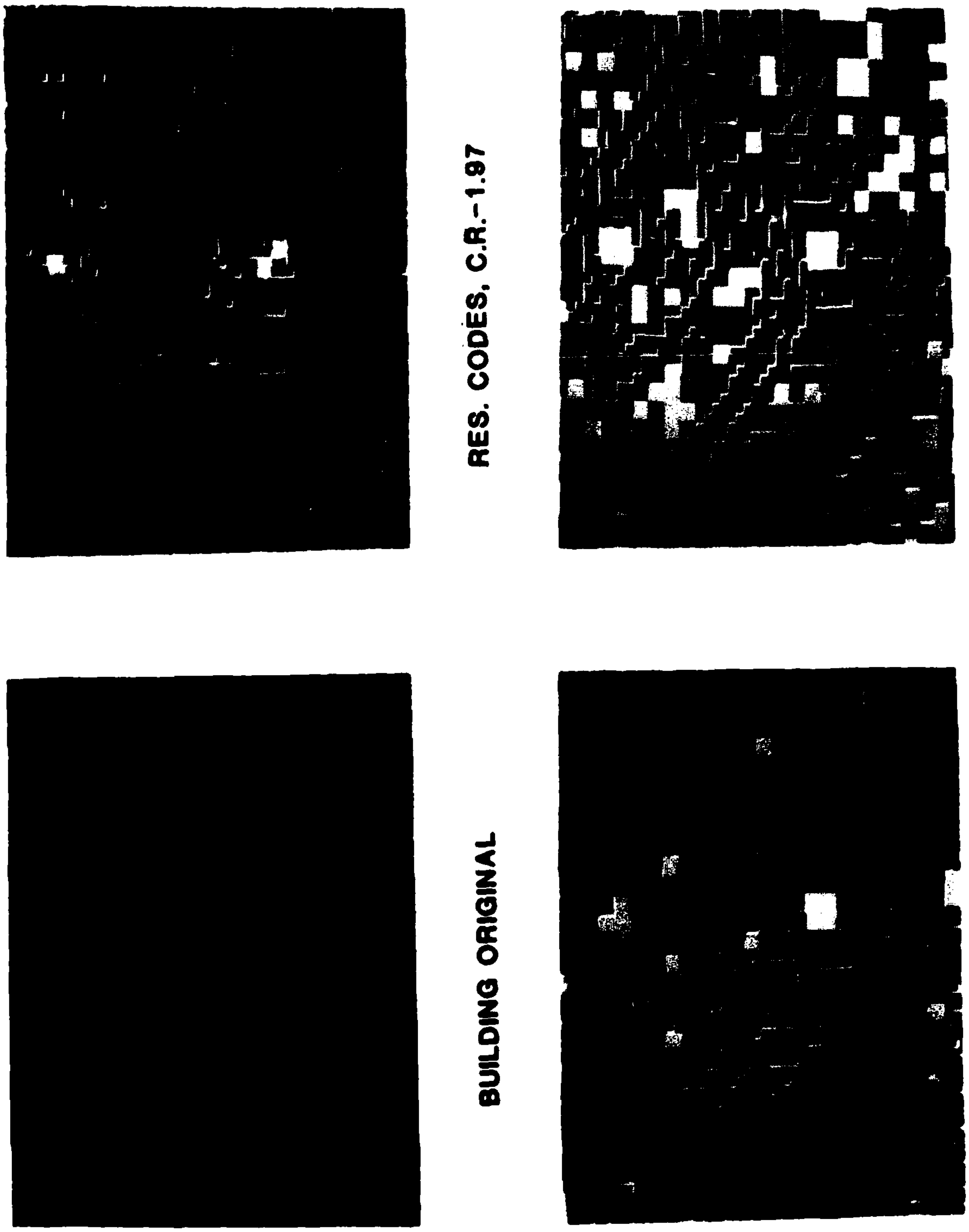


Figure 5-6 $\begin{pmatrix} E & F \\ G & H \end{pmatrix}$ Varying Compression Ratios (continued)

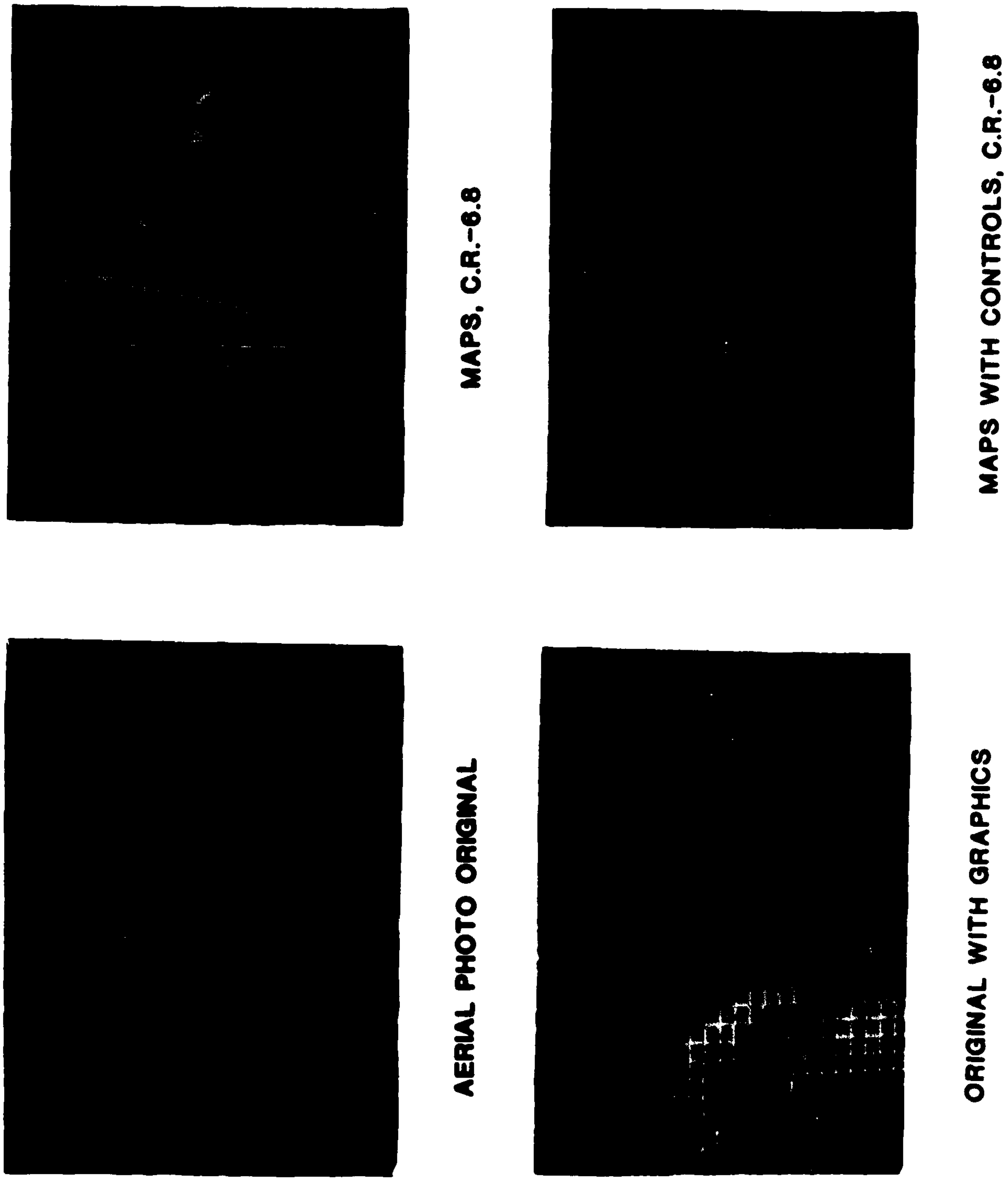


Figure 5-7. $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ Interactive Macro-Fidelity Examples with Zoom and Split Screen

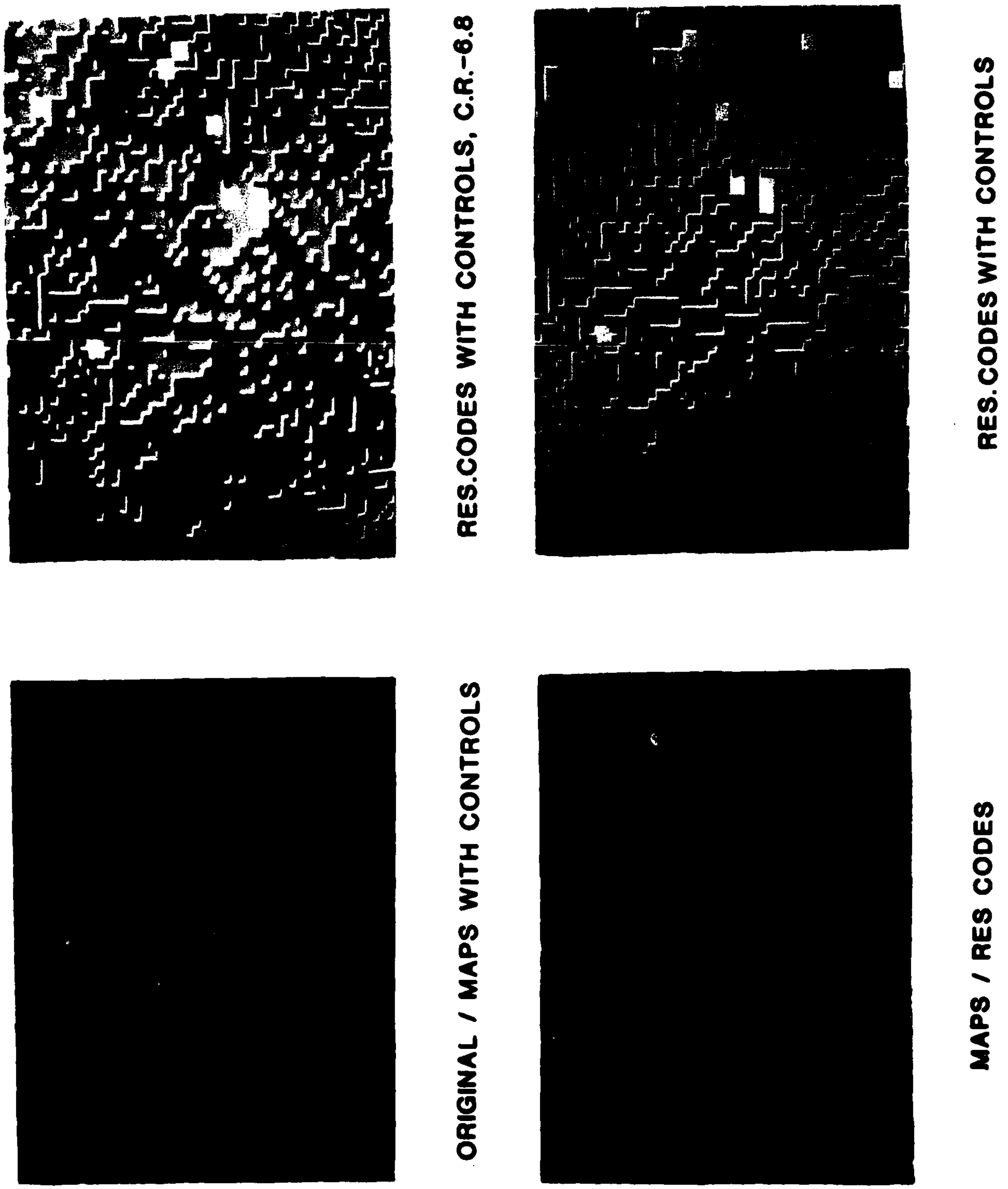


Figure 5-7. $\begin{pmatrix} E & F \\ G & H \end{pmatrix}$ Interactive Macro-Fidelity (continued)

SECTION SIX
RECOMMENDATIONS — TRANSPORTABLE SOFTWARE

Implementation of MAPS in microcode on various high-speed image processing systems will be critically dependent on the architecture, instruction set, and capabilities of each particular system. However, the generic considerations described in section 5.2 and the special consideration outlined in Appendix B should provide useful guidance.

For more general familiarization and exploration of the MAPS technique, however, preparation of a "transportable software" version written in a higher-order language (probably FORTRAN) is recommended. The basic MAPS concepts are sufficiently mature to justify such formalization and sufficient options should be included to provide insight into the scope and diversity of the technique. An outline of potential options for inclusion in the implementation is included as Figure 6-1.

MAPS is inherently integer and arithmetically simple in organization. The principal barriers to complete transportability thus reside in interactions with word size, memory capacity, and file structure in the host machine. For this reason, input and output operations and pixel decode/MAPSel encode operations should be broken out as modular routines. Any specific differences of a particular system from the target machine will then be largely localized to these areas. Also impacting these areas is the requirement that the source and MAPS imagery have selectable intensity code lengths of six or eight bits.

MAPS OPTIONS

INPUT/OUTPUT; PIXEL DECODE (6 OR 8 BIT)

IMAGE SIZE/SECTION SELECTION

MULTI-PASS/MULTI-IMAGE PROCESSING

MODE SELECTION: COMPRESSION ONLY
COMPRESSION/BLOCK DECOMPRESSION
COMPRESSION/ADAPTIVE DECOMPRESSION
BLOCK DECOMPRESSION ONLY
ADAPTIVE DECOMPRESSION ONLY
"LEVEL" IMAGE DECOMPRESSION
DIFFERENCE IMAGE (SIGNED, AMPLIFIED)

MAPS SUBFRAME SELECTION: 8x8 OR 16x16

REARC/SELECTABLE MACRO-FIDELITY CONTROL

MICRO-FIDELITY CONTROL SPECIFICATION: CONTROL MATRIX
MATRIX GENERATORS
CONTRAST SCALE ONLY

INTENSITY PARTITION/RESET OPTION: MEAN
PSEUDO MEDIAN
BIASED MEDIAN
EXTREME

ADAPTIVE DECOMPRESSION CONTROL: CONVOLUTION WEIGHTING
LEVEL ACTIVATION
INTENSITY ACTIVATION
SHARPENING
DITHER

IMAGE LABELLING: TAPE
MAPS ANNOTATION

PERFORMANCE SUMMARYS: COMPRESSION (RATIO, BITS/PIXEL)
LEVEL HISTOGRAMS (% OF IMAGE)
INTENSITY HISTOGRAMS (2-D)
ERRORS - RMS, MSE, RMSE

Figure 6-1. Potential User Options, MAPS Transportable Software

Image size should be user selectable along with the ability to "section" the image by skipping in a specified number of pixels and lines. In addition, the user should be able to achieve multiple passes through the same image at varying control parameters or process multiple images in sequence on the source file.

A variety of input/output modes should be available. Among these are creation of a file containing a MAPS image in compressed form, capability for decompressing such an image in either the BLOCK or ADAPTIVE mode, or compression followed immediately by decompression in either tonal image mode. Generation of various related images is also desirable; for example, a "level" image where each resolution code is displayed as a selected gray scale and difference or error images. The latter may be "signed" differences biased by a neutral middle gray value or "amplified" absolute differences.

In the area of MAPS geometric control, the user should have the option of choosing 8x8 or 16x16 subframes and of designating various subregions for alternate fidelity control with a REARC-like (REdundant ARea Coding) coding scheme.

Micro-fidelity control options should include specification of the contrast control matrix directly, generation of the matrix from four parameters, or generation from the contrast scale alone with the other three parameters set to default values ($B=3$, $F=0.5$, $\Delta=0.1$). In addition to the spatial partition of the image for macro-fidelity variation, an intensity partition capability with various element replacement strategies and alternate fidelity controls should be included.

For the ad .ive decompression mode, the user should be able to supply a weighting function or select from several internally available, and to place constraints on which surround elements will be activated according to their resolution level and intensity compared with the target element.

Moreover, a selectable "dither" level to mask contouring should be incorporated.

Two types of labelling should be supplied. The first is a file header record which gives the name and size of the image. the mode (compressed, decompressed, or evaluation), and other control options exercised. The second should allow for alphanumerics to be overlaid directly on the image bit stream for hard or soft copy visual annotation.

Finally, a variety of performance summaries related to intensity and resolution distributions, compression levels, and intensity errors should be reported as part of the output listing.

A software package with these features would provide services ranging from those of a basic tool for further research to a user capability for compact storage of a significant data base of digital imagery.

Appendix A

A.0 FLEXIBLE PROCESSOR AND SCAN CONVERTER MEMORY

CONTROL DATA FLEXIBLE PROCESSOR

The Control Data Flexible Processor is a digital microprogrammable processor which has been designed specifically for multidimensional array processing. Since the units are microprogrammable they combine the advantages of special-purpose hardware in efficiency and speed of software changes to the computational algorithms.

Each Flexible Processor can be used as a building block in a modular system design where each Flexible Processor is accordingly programmed with appropriate algorithms to perform specific tasks which are assigned to it. Moreover, programming of the various Flexible Processors in an array can be changed dynamically under control of a processing system which calls prescribed programs from memory, as from a rotating disk file. The size of the Flexible Processors has been effectively chosen to permit efficient construction of a complete system tailored to meet project requirements.

The unique features of the Flexible Processor make it ideally suited for systems applications in a wide variety of scientific disciplines. One such area of application is digital image processing. Typical computational functions which can be solved with this processor include image matching, correlation, spatial transformation, registration, radiometric corrections, change detection, statistical classification, and various enhancements. Applications in the field of image processing include cartography, reconnaissance, management of earth resources, non-destructive testing, and biotechnology. Additionally, the Flexible Processor is ideally suited for control of image display and recording equipment.

The paragraphs which follow describe the technical characteristics, programming, and uses of the Flexible Processor.

1.0 HARDWARE AND SOFTWARE DESCRIPTION

The Flexible Processor is a special-purpose computing unit featuring high arithmetic computation rates, considerable character-handling capabilities, an advanced input-output structure and semiconductor register file memories for data storage. Each unit features arithmetic logic capable of a 0.125 microsecond addition of 16-bit or 32-bit operands, a 0.250 microsecond fixed point multiply of 8-bit bytes, and a 1.125 microsecond fixed point multiply of 16-bit operands (TABLE I).

TABLE I. CDC FLEXIBLE PROCESSOR CHARACTERISTICS

- MICROPROGRAMMABLE – RANDOM ACCESS MICROCONTROL MEMORY
- 32-BIT OR 16-BIT WORD LENGTHS
- ARRAY HARDWARE MULTIPLIER
- 16 LEVEL HARDWARE PRIORITY INTERRUPT MECHANISM – 3 LEVEL MASK CAPABILITY
- SPECIALIZED LOGIC FOR SQUARE ROOT AND DIVIDE
- 8 mHz FILE BUFFERED WORD TRANSFER RATE – 16 WORD BY 32-BIT OR 16-BIT INPUT FILE BUFFER
- 2 mHz DIRECT MEMORY ACCESS WORD TRANSFER RATE
- 1 mHz REGISTER-BUFFERED WORD TRANSFER RATES
- DUAL 16-BIT INTERNAL DATA BUS SYSTEM
- 0.125 μs CLOCK CYCLE
- 0.125 μs 32-BIT ADDITION: 0.250 μs BYTE MULTIPLICATION
- REGISTER FILE CAPACITY UP TO 4128 SIXTEEN-BIT WORDS
- HARDWARE NETWORK FOR CONDITIONAL MICROINSTRUCTION EXECUTION – 4 MASK REGISTERS AND A CONDITION HOLD REGISTER

The Flexible Processor is a dual bus organized machine with a modular set of functional units centered around a control section including conditional instruction execution networks, interrupt hardware, and featuring a read/write semiconductor micro-memory (Figure 1). The micro-memory word contains both direct control functions and decoded bus controls. Movement of operands among the functional units is particularly effective in applications which require a high data throughput, as in image processing.

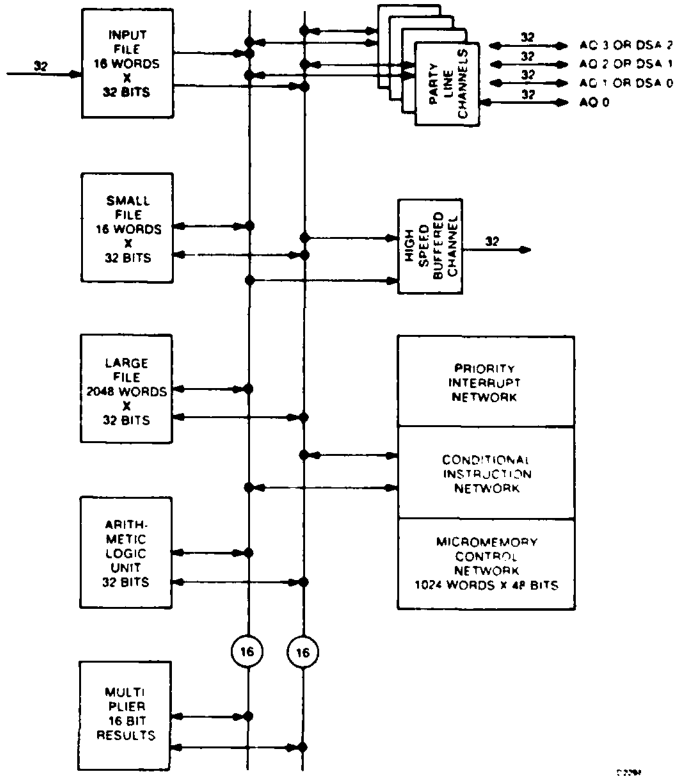


FIGURE 1. DATA PATH ORGANIZATION IN THE CDC FLEXIBLE PROCESSOR

1.1 Functional Units

Three basic functions comprise the operation of the Flexible Processor: Arithmetic, Register File Storage and Input/Output. These are described as follows.

1.1.1 Arithmetic

Arithmetic functional units are of two kinds: Arithmetic Logic Unit (ALU) and Hardware Array Multiplier.

Arithmetic Logic Unit

This unit is capable of thirty-nine unique arithmetic and logic operations. Shifting of all input feeder registers, *right or left*, with a selection of inputs to the registers during the shift instruction, and a "scale" capability on the output of the ALU extend the capabilities of this functional unit. The ALU may be configured in a 16-bit or 32-bit format (one and two cards, respectively).

Hardware Array Multiplier

The Hardware Array Multiplier is a modular arithmetic function which is capable of eight by eight multiples with each eight bit operand directly selectable from two sixteen bit input registers. Multiplication is performed asynchronously to the control unit, allowing other instructions to be executed while the multiplication is taking place.

1.1.2 Register File Storage

The internal Register File Storage is semiconductor memory providing both Temporary and Large File functions (TABLE II). The Temporary File provides 16 words of 16 bits each for each bus. This file has separate read and write address registers, allowing simultaneous read and write capability. The Large File

TABLE II. FILE STORAGE

TYPE	DESCRIPTION	OPTIONS NUMBER OF WORDS X 16 BITS
TEMP FILE	60 ns semiconductor Simultaneous Read and Write 16 bits per chip 1 clock cycle per access	2-16 word files 1-16 word file none
LARGE FILE	60 ns semiconductor Read/Write 256 or 1024 Bits per Chip 1 clock cycle per access	2-2048 word files 1-2048 word file 2-1280 word files 1-1280 word file 2-1024 word files 1-1024 word file 2-512 word files 1-512 word file 2-256 word files 1-256 word file none

provides a maximum storage of 2048 words of 16 bits each for each of the two buses.

1.1.3 Input/Output

The Flexible Processor has three types of external data transmission functional units available. These are:

- Bi-Directional Party-Line Channel (AQ)

This channel is compatible with CDC 1700 peripheral equipment and has a maximum data rate capability of one megaword per second (16 bit data word). There is a maximum capacity of up to 4 of these channels; however, they share space with the DSA channel.

- External Mass Memory Access Channel (DSA)

This channel is primarily intended for use with an MOS semiconductor mass memory. This channel has a data rate capability of two megawords per second (16 bit data word). The maximum addressable memory contains over one million sixteen bit words (20 address bits, four of which are used for bank selection). There is a maximum of three of these channels; however, one channel is forfeited for each additional AQ channel in the Processor.

- Inter-Flexible Processor Communication Channel

This functional unit permits asynchronous data transmission between Flexible Processors at 8 megawords per second with a maximum word length of 32 bits. In terms of the bit transfer rate, the maximum communication rate between Flexible Processors is 256 million bits per second. This communication channel is comprised of two dependent sections. One section is located in the sending Flexible Processor, and the other section is in the receiving Flexible Processor. The receiving section has a sixteen word memory capable of simultaneous read and write operations enabling transmission asynchronous to the receiving Flexible Processor's usage of the operands (TABLE III). Each Flexible Processor has a maximum capacity of two 16 bit word sending sections and two 16 bit word receiving sections. In addition, each sending section can communicate with as many as four receiving sections

1.2 Miscellaneous Hardware Capabilities

Miscellaneous hardware capabilities of the Flexible Processor are Loop Counters, Return Jump File, Micromemory Control Storage, and optional External Memory. These capabilities are described in the following paragraphs.

1.2.1 Loop Counters

Four counters are provided for software problems involving iterative computation. There are three 8-bit counters and one 16-bit counter with two compare registers associated with each counter.

1.2.2 Return Jump File

A return jump file of sixteen words is used for subroutine exits, interrupt returns, and iterative loop returns. This is a push down-push up file capable of returning control to a main program through a maximum of sixteen nested loops or subroutines.

1.2.3 Micromemory Control Storage

The micromemory consists of 48-bit words of random access read and write semiconductor memory with increments from 256 words to 1024 words for program storage.

1.2.4 External Memory

A MOS semiconductor memory bank can be added to a basic Flexible Processor by adding a DSA channel in the Flexible Processor and one or more racks of memory (Figure 2). This external memory is expandable in 8192 word increments from 8192 sixteen-bit words to 65536 sixteen-bit words per bank with a maximum of sixteen banks per DSA channel. This memory has the capability of multiport operations with any one port having a data rate capability of two megawords per second and multiport operation of three megawords per second. Faster memory capabilities for any Flexible Processor can be achieved by three DSA channels in the Flexible Processor communicating with separate ports of unique banks of memory.

TABLE III. INPUT STORAGE

TYPE	DESCRIPTION	OPTIONS NUMBER OF WORDS BY 16 BITS
Input File	• 60 ns semiconductor • Simultaneous Read and Write • 16 bits per chip • 1 clock cycle per access	2-16 word files 1-16 word file none

1.3 Software

As stated previously, the Flexible Processor is a microprogrammable processor with read and write random access instruction memory. Thus, through this memory the Flexible Processor is encoded to compute a desired algorithm, and the memory can be rewritten subsequently to perform different algorithms if it is so desired.

The instruction is a 48-bit word containing both decoded and direct control fields. The instruction word defines the following operations:

- Simultaneous control of the Arithmetic Logic Unit, Multiplier, Jump Stack, Loop Counter, Conditional Execution, and the two buses.
- Independent shifting of three Registers.
- Simultaneous control of four Input/Output Channels.

A cross assembler, MPASS2, has been written to aid in the generation of microprograms for the Flexible Processor. This assembler runs on the CDC 1700 computer system utilizing a data deck that has a one-to-one correspondence between the micromnemonics and the microinstruction control fields. A symbolic addressing capability of the assembler also eases the writing of microprograms.

Other available software routines are:

- Library routines for generation of a microprogram library on disk file.
- Loader routine for loading microprograms into the Flexible Processor from the library disk file.
- Diagnostics for all functional units in the Flexible Processor.
- Display Station Monitor for hardware and software checkout of the Flexible Processor and microprograms.

1.4 Physical Description

Another attractive feature of the Flexible Processor is that site requirements are minimal. The mechanical configuration, cooling requirements and power requirements are described in the following paragraphs.

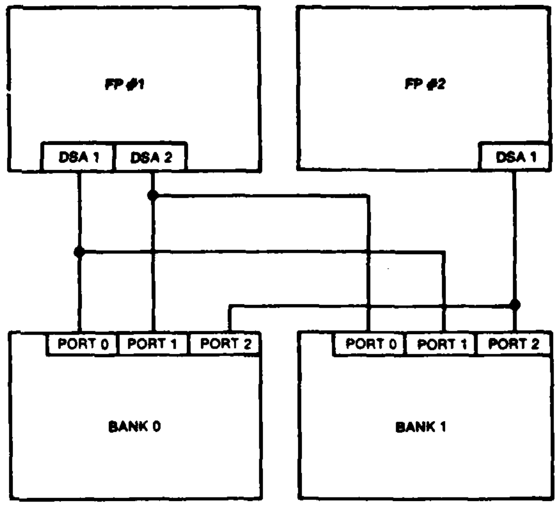


FIGURE 2. MULTIPLE-PORT DUAL-BANK EXTERNAL MEMORY

1.4.1 Mechanical Configuration

The Flexible Processor functional units and control unit are contained within a card cage that fits into a standard 19 inch relay rack. The dimensions of the card cage are:

- Length: 19 inches (482 millimeters)
- Height: 7.883 inches (200 millimeters)
- Depth: 12 inches (305 millimeters)

The card cage contains forty-one 120 pin edge-board connectors that have a standard mechanical and wire wrap configuration for any Flexible Processor.

1.4.2 Cooling Requirements

The Flexible Processor is forced-air cooled. Each card module location of the Flexible Processor shall receive 450 lineal feet (117 meters) per minute of forced air entering the FP from below the logic rack through the card guides.

1.4.3 Power Requirements

The power requirements for the Flexible Processor in a basic configuration are typically:

- +5VDC ± 5% @ 45 amperes
- —5VDC ± 5% @ 3 amperes

2.0 PIPE-LINE AND OFF-LOADING PROCESSING

Use of the Flexible Processor falls into two major system philosophies:

- Pipe-line processing
- Off-loading from a general-purpose computer

The Flexible Processor was specifically designed to process image data in a pipe-line mode, thus making possible very high data rates. In this system concept the process flow involving the data proceeds directly through a special purpose processor constructed of an array of Flexible Processors (Figure 3). Process control is effected with a general purpose computer or a special controller constructed of Flexible Processors.

In other instances it is convenient to off-load burdensome computations from a general purpose computer to an array of Flexible Processors (Figure 4). This is done where it is desired to perform computation on the data in a general-purpose computer, but it is found that certain computations are particularly burdensome to the central processor. Then total system performance can be greatly improved by off-loading these computations to an array of Flexible Processors.

As was described previously, each Flexible Processor can communicate with another Flexible Processor, with a host computer, and with external memory. This is illustrated in Figure 5 for a standard packaging of 4 Flexible Processors and 144K of external MOS memory. The Flexible Processors can operate in series or parallel combination to meet the process requirements. Notation in Figure 5 is as follows:

FP — Flexible Processor

IF — Input File

OR — Output Register

AQ— Party Line Transmission Channel

DSA — External Memory Access Transmission Channel

MOS — Random Access Memory

ST — Station

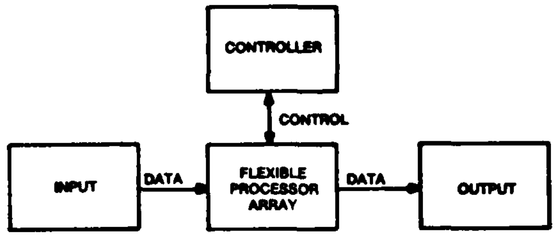


FIGURE 3. PIPELINE PROCESSOR CONFIGURATION

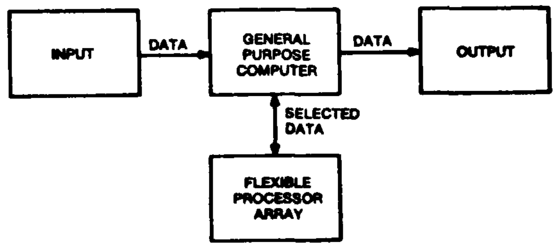


FIGURE 4. OFF-LOADING PROCESSOR CONFIGURATION

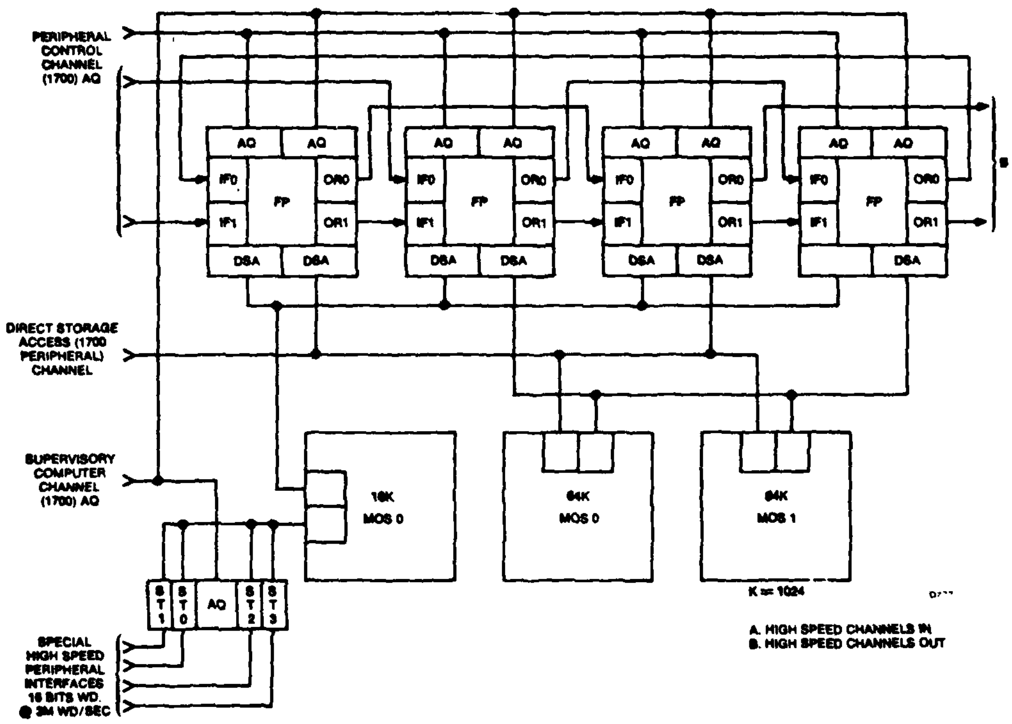


Figure 5. Configuration of 4 CDC Flexible Processors and Augmented Memory

3.0 CONTROLLER AND PROCESSOR FUNCTIONS

Uses of the Flexible Processor in image processing applications can be divided broadly into the following three categories:

- Preprocessor
- Applications Processor
- Digital Controller for Image Display and Recording Equipment

In each of these areas many benefits accrue from processing the image data in digital form. A precision and complexity of manipulations can be attained that is difficult, if not sometimes impossible, to achieve in strictly analog fashion. Flexible Processors have been used for image matching, correlation, spatial transformation, radiometric correction, and enhancement. The built-in memories together with additional memory are conveniently used for table look-up functions, which might include data corrections, symbol generation, and alpha-numeric generation. Any of these items may appear as a preprocessing requirement in an image processing application.

The Applications Processor is here defined as that processor which performs interpretive functions upon the data. Typical examples are change detection, statistical classification, and stereo mapping (TABLE IV). Many of the Applications Functions may be performed automatically, while others may be computed as a result of operator interaction.

In many instances data from a number of sources must be coordinated at an image display or recording station. To maintain fidelity of the data it is advantageous to use a digital controller at the display or image recorder, and the Flexible Processor is ideal for such use (TABLE V).

TABLE IV. USE OF FLEXIBLE PROCESSORS FOR APPLICATIONS PROCESSING

- CHANGE DETECTION
- CARTOGRAPHY
- STATISTICAL CLASSIFICATION
- TARGET RECOGNITION
- TACTICAL DEPLOYMENT CONCENTRATION
- MOVEMENT VECTORS
- IMAGE CORRELATION
- SUPERPOSITION AND REGISTRATION
- GEOMETRIC CORRECTION
- RADIOMETRIC CORRECTION

TABLE V. USE OF FLEXIBLE PROCESSORS IN IMAGE DISPLAY CONTROLLERS

- SCAN CONVERSION
- ROUTINE IMAGERY CORRECTION WITH ANCILLARY DATA
- GENERATION AND PLACEMENT OF IDENTIFYING TARGET SYMBOLS
- GENERATION OF ALPHANUMERIC INFORMATION
- IMAGE SCENE SELECTION
- ENHANCEMENT MODES
- TIMING AND CONTROL FUNCTIONS FOR DISPLAY EQUIPMENT
- GENERATION OF IMAGE COMPOSITES

4.0 RELATED EXAMPLES

Three examples are described here to illustrate the versatility and data handling capacity of Flexible Processor configurations. The applications described are:

- Digital Change Detection
- Preprocessor for LANDSAT (ERTS) Data
- Digital Scan Converter Memory System

The following examples demonstrate the capability of Flexible Processor systems to perform complex processing functions at high through-put data rates. These systems are cost effective and therefore deserve attention in similar applications.

4.1 Digital Change Detection

The 4-Channel Modular Change Detection System performs automatic digital change detection at a *nominal processing rate of* 830,000 pixels per second from each of two input images. Each pixel* is encoded to 8 bits, hence the average processing rate exceeds 13 million bits per second. The system was developed in a program with the Rome Air Development Center.

Extraordinary performance, cost-effectiveness, and growth potential are achieved with modular architecture incorporating an array of forty Control Data Flexible Processors, each of which is micro-programmable. Each Flexible Processor is microcoded to perform a specific computational algorithm which is assigned to it. High speed interprocessor communication and data transfer permit the use of combined parallel and serial configurations yielding optimum system performance.

The system is designed to perform change detection in real time. The two *data sets are matched and correlated* in a totally digital mode. Then the mission image data is spatially transformed (digitally) to register precisely with a reference image. A difference image is created by subtracting gray-scale values (tonal values) of the mission image from the reference image on a point-by-point basis.

*A pixel is a picture cell or image sample element.

This difference image is further enhanced to emphasize difference detail and to remove noise. A unique feature is false alarm suppression with a feature-oriented processing technique.

Processing is carried out in four *Change Detection Processors* (Figure 6), each of which is assigned to one of four image channels. There are 9 Flexible Processors per channel programmed variously for processing algorithms, 2 Flexible Processors for output control, and 2 Flexible Processors for output overhead (a total of 40 Flexible Processors). A *large digital buffer* stores 9 million bits of mission image data.

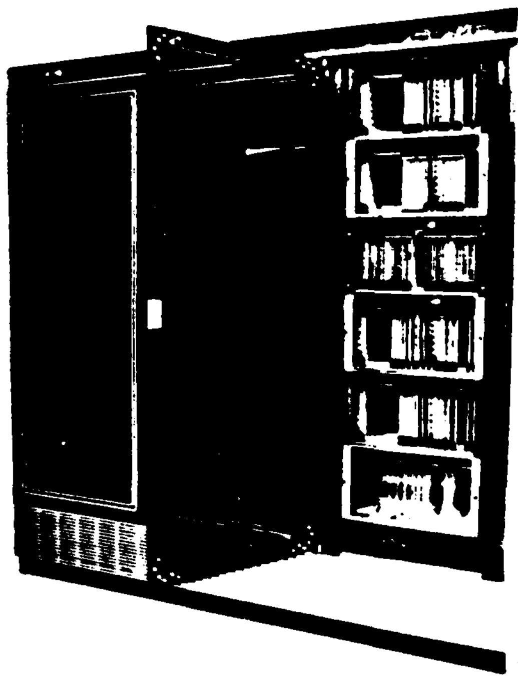


Figure 6. Change Detection Processor

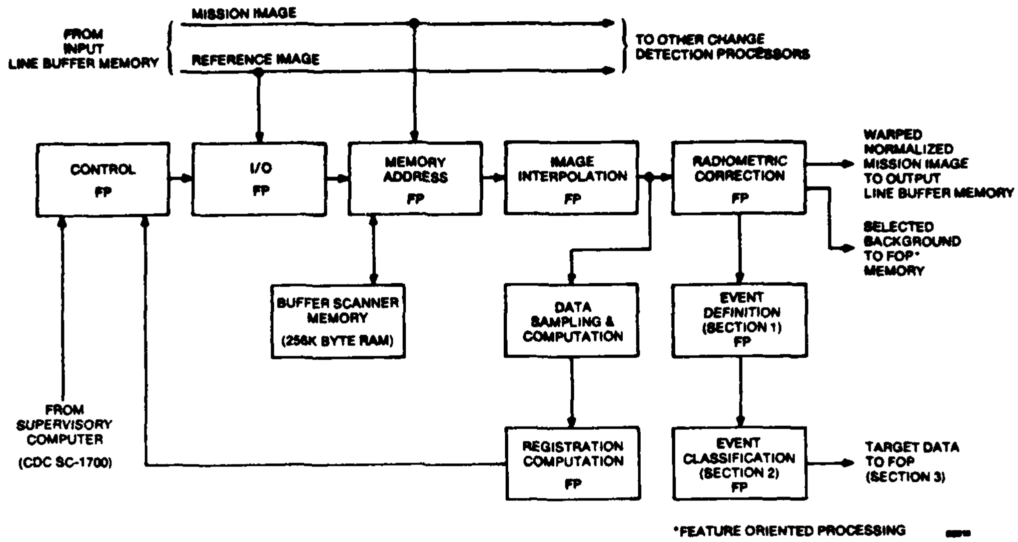


FIGURE 7. BLOCK DIAGRAM OF CHANGE DETECTION PROCESSOR (SINGLE CHANNEL)

Processing of image data proceeds in pipeline fashion from the input source, through the various sections of the Change Detection Processor, and to output devices (Figure 7). Relative image distortions are corrected via a feedback control loop *which operates on the memory* address. The process is supervised with a Control Data SC-1700 Computer. A library disk provides diagnostic and real-time algorithm firmware.

Original scene and difference image data can be written on any one of three 9-track high speed digital tape drives, operating at 8000 BPI. These tape units can also be used as input data sources to the system.

4.2 Preprocessor for LANDSAT (ERTS) Data

An example of the use of Flexible Processors in preprocessing of LANDSAT (ERTS) data is the Terra Information Processing System (TIPS) constructed for Telespazio in Italy. This system will process LANDSAT and airborne remote sensing data as the prime application.

TIPS converts satellite image data to computer compatible tape in real time, provides both visual and automatic quality control checks during the process, and provides a capability for subsequent viewing, editing, and analysis of the imagery, with final output on either computer-compatible tape or film transparencies. A unique feature of the system is the Flexible Processor which performs the basic corrections and formatting in real time directly from the satellite telemetry.

The TIPS system uses a high performance *configuration for central* control (Figure 8). A wide range of input data formats, rates, and preprocessing functions is possible with the use of two Flexible Processors at the front end. This design configuration provides maximum utilization of standard proven electronic data processing equipment, and the very high data processing rates are achieved through the use of the microprocessors. It is significant that

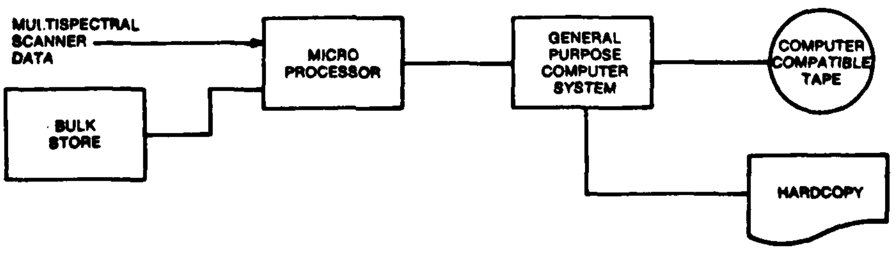


Figure 8. Terra Information Processing System (TIPS)

the microprocessors may be reprogrammed for a wide range of tasks in addition to the processing of digital image data.

4.3 Digital Scan Converter Memory System

A Digital Scan Converter Memory System provides refresh for a 3-color video display system at an average rate of 78 million bits per second. The image frame consists of 512 lines of 640 pixels per line with each pixel encoded to 8 bits. The image is line interleaved and is refreshed at 60 fields per second or 30 frames per second. The system configuration (Figure 9) includes a CDC Flexible Processor and external memory which performs the following primary functions:

- Perform controller functions such as memory address computation, load gray-scale table look-up memory, etc.
- Load and/or modify contents of the scan converter memory
- Translate operator inserted keyboard commands into alphanumeric and graphics overlay data.

The capacity of the scan converter memory is 640 pixels per line by 512 lines per frame by 8 bits per pixel. The memory is organized into four banks each having a capacity of 40K words (K=1024) by 16 bits per word with two eight bit pixels packed into each word. The number of banks can be modularly expandable up to a maximum of 16 banks. Thus four independent 640 x 512 x 8 digital scan converters could be controlled by one Flexible Processor. With some modification, this memory capacity could be extended to a single high resolution display of 1280 x 1024 x 8. In addition, each bank is expandable to a maximum capacity of 64K x 16. Thus, with 16 banks the maximum capacity is 1048576 words (16-bits each).

Since the memory is random access, it may also serve a dual role as an extended capacity random access storage for data processing calculations. The memory cycle time for either read or write is 300 nanoseconds.

Keyboard and trackball or light pen inputs are translated by the Flexible Processor so that memory contents can be appropriately modified to show an updated cursor position on the CRT display. This allows image pixel interrogation with the cursor which is especially useful for image analysis and interactive editing.

Image zooming and offsetting is implemented by appropriate keyboard entry into the Flexible Processor.

Level slicing and image feature enhancement is implemented by a table-look-up memory.

Scrolling of continuous strip images into a "moving window" format from the CRT display can be accommodated by keyboard call of an appropriate FP microprogram.

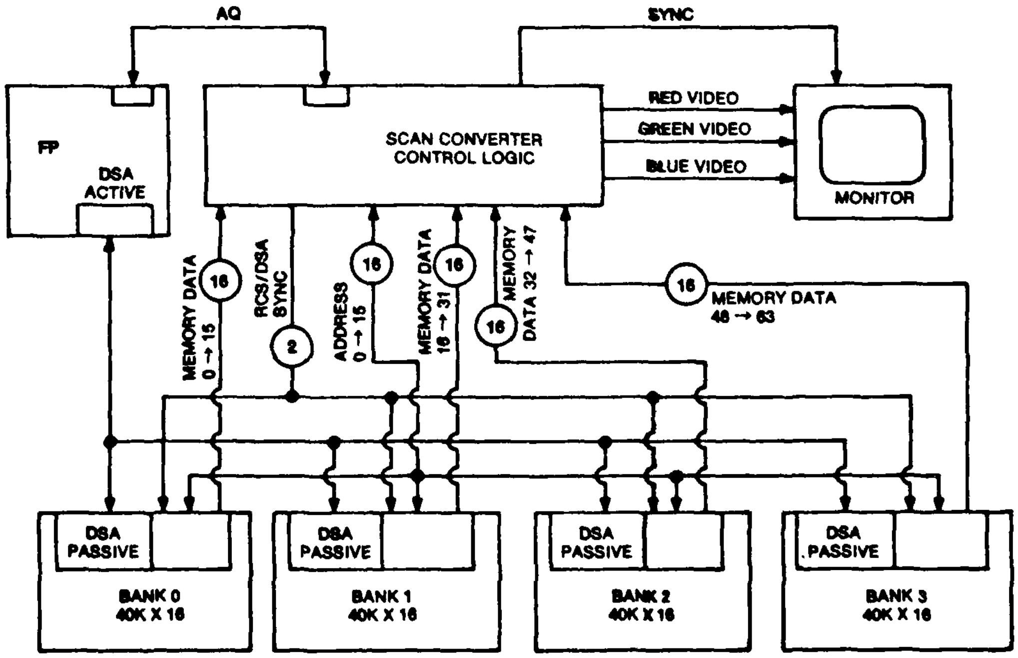


Figure 9. Digital Scan Converter Memory System

Appendix B

B.0 SPECIAL CODING CONSIDERATIONS

This section shows the micromemory and large file allocation for the MAPS macrofidelity control program. Figures are included to show the derivation of the MAPS ordering lookup table and the sorting lookup table. A glossary of table and routine names is also included.

B.1 Micromemory and Large File Usage

The entire MAPS overlay requires 710 ($AC8) words of micromemory. *It is an overlay to the DCON* program and requires several* DCON routines in order to function. A map of micromemory is shown in figure B-1. Approximately 2/3 of the micromemory required is used to support the interactive capabilities and the remainder holds the compression and decompression routines. The overlay requires 310 ($136) words of large file 0 and 303 ($52F) words of large file 1. Figures B-2 and B-3 show a map of large file usage. Additionally the overlay uses the Character Generator Parameter Tables (CGPT) in DCON's area of large file. A complete breakdown of requirement on a routine basis is shown in figure B-4.

*DCON — Display Controller

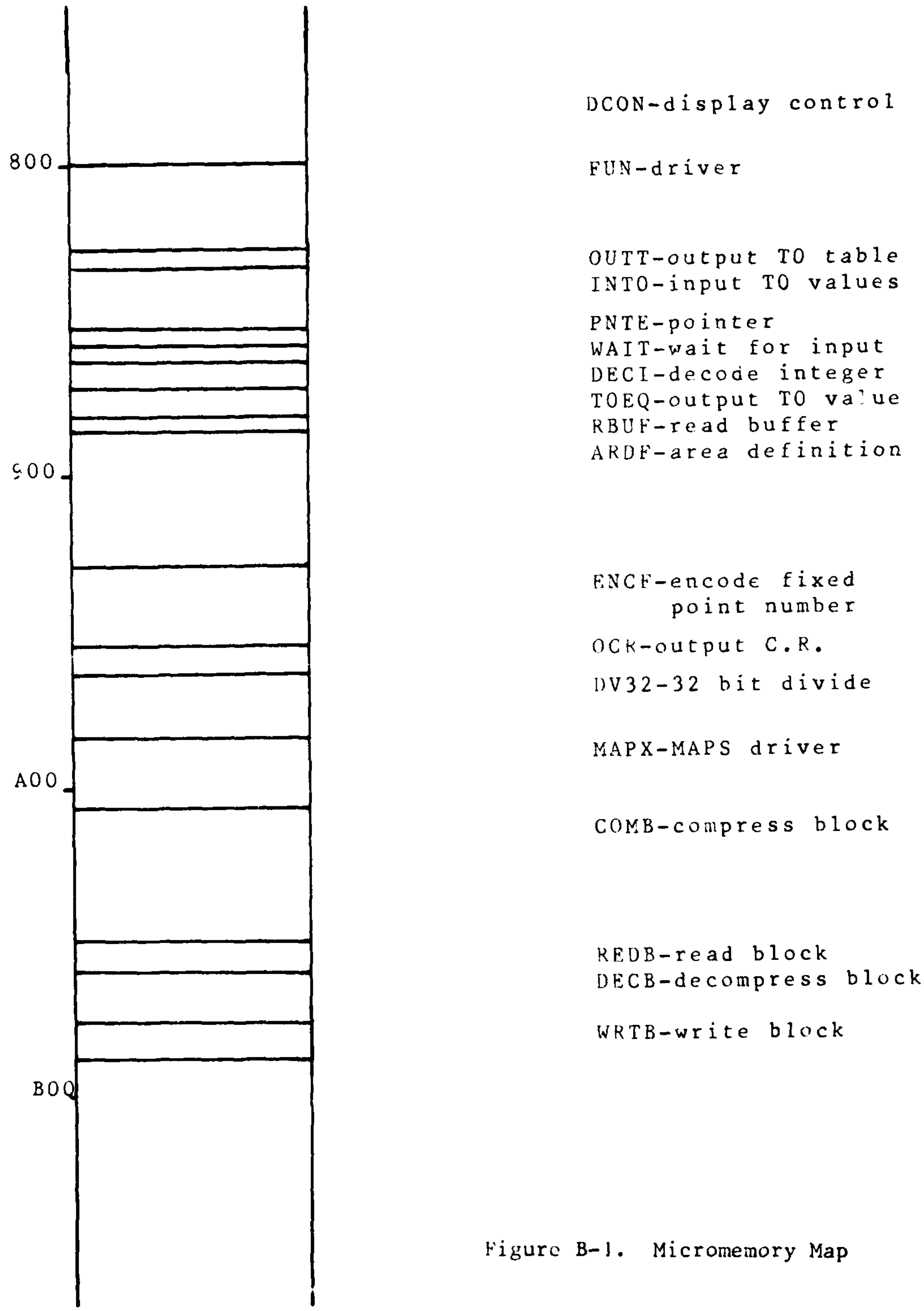
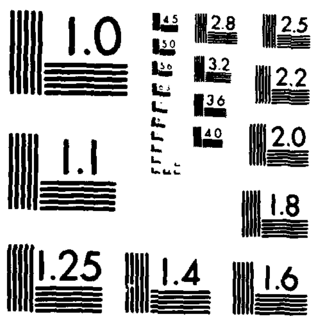


Figure B-1. Micromemory Map

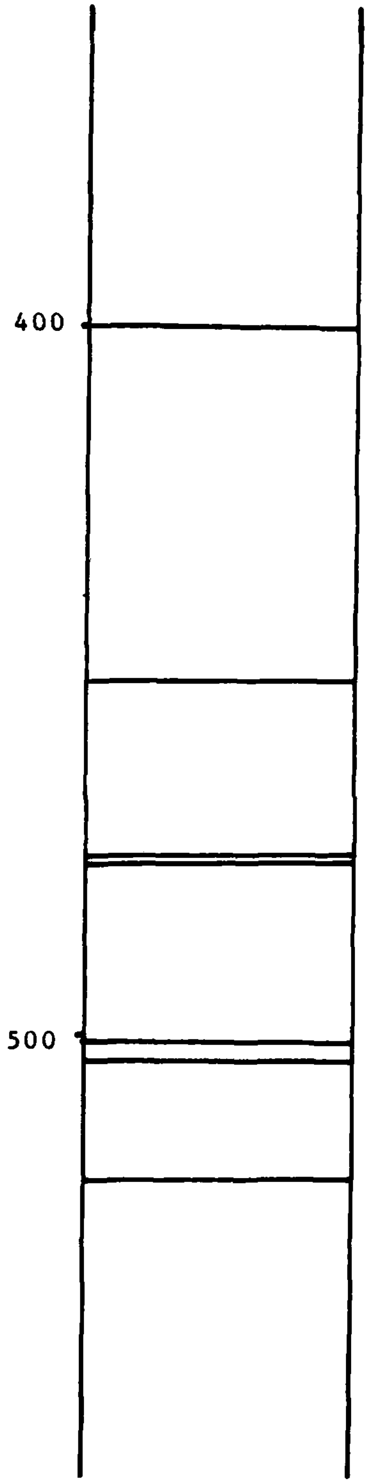
AD-A088 763 CONTROL DATA CORP MINNEAPOLIS MN IMAGE SYSTEMS DIV F/G 5/8
MAPS IMAGE COMPRESSION.(U)
MAY 80 A E LABONTE, T E ROSENTHAL F30602-78-C-0253
UNCLASSIFIED ISD-78003-80-R1 RADC-TR-80-173 NL

2 - 2

END
DATE FILMED
10 80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A



DCON-display control

400 MAPS ordering look-up table

481 sorting lookup table

4C1 COMB indirect addr.
4C3 TO(0) CCM
4D3 TO(1) CCM
4E3 TO(2) CCM
4F3 TO(3) CCM

503 ARDF state table
50A MES1 message buffers
510 MES2
519 MES3
524 MES4
52F MES5

Figure B-2. Large File 0 Map

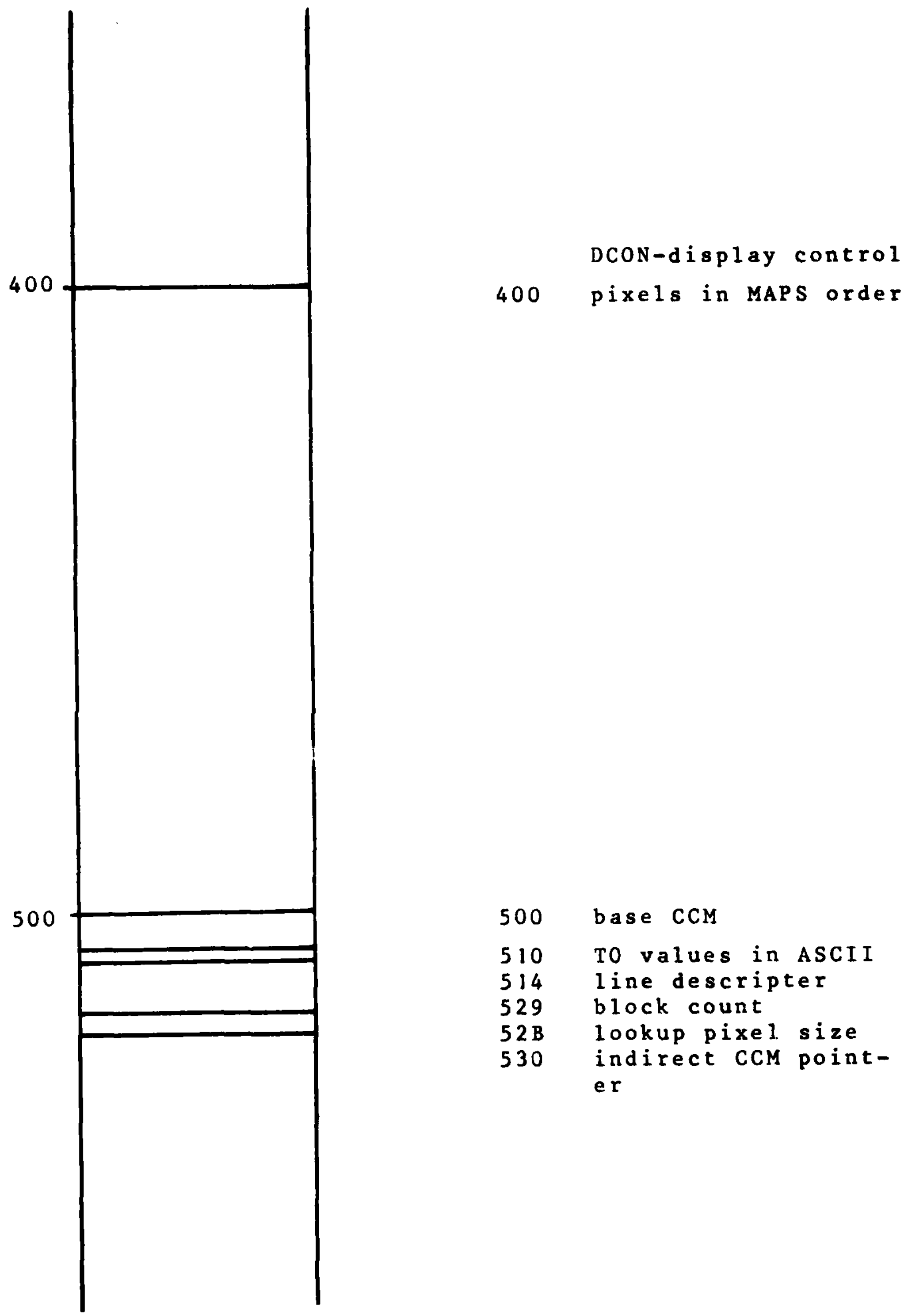


Figure B-3. Large File 1 Map

Name	Location	Large File 0 Usage	Large File 1 usage	Routines called
FUN	802-849	Schedule 380-385 MESS 52F-535	HSCG 3DO-3DF MYCG 3EO-3EF	OUTT, INTO, ARDF COF1, MAPS, CON1, OCR, DBUF, WAIT, CLR, CLRI, CGO1
OUTT	84A-853			TOEO
INTO	854-881	MES2 510-519 CCMS 4C3-502	ASCI 510-513 BASE 500-50F	WAIT, DECI, PNTE, MSP, TOEQ, CGO1
PNTE	882-893		MYCG 3EO-3EF	CGO1
WAIT	894-8A5		HSCG 3DO-3DF	CLR
DECI	8A6-8BA			RBUF
TOEQ	8BB-8DA	MES1 50A-510	MYCG 3EO-3EF ASCI 510-513	DBUF
RBUF	8D3-8DD			
ARDF	8DE-949	STAT 503-507 MES3 519-523	HSCG 3EO-3EF FRAM 514-52F	PNTE, CLR, DBUF, DRAW
ENCF	94A-98A			WBUF
OCR	98B-9A3	ME34 524-52E BCNT 4C2-4C3	MYCB 3EO-3EF	DV32, DBUF, ENCF
DV32	9A4-9D8			
MAPS	9D9-AOF	CCMS 4C3-502	Indirect 530	REDB, COMB, DECB, WRTB
COMB	A10-A80	SORT 481-4CO CCMS 4C3-502 Indirect 4C1	Pixels 400-4FF Pixel size 52B-52F Indirect 530	
REDB	A81-A9E	MAPO 400-480	Pixels 400-4FF	
DECB	A9F-AC2	BCNT 4C2-4C3	Pixels 400-4FF Pixel size 52B-52F	
WRTB	AC4-ADB	MAPO 400-480	Pixels 400-4FF	

Figure B-4. Detailed Breakdown of Memory Allocation

B.2 MAPS Ordering Look Up Table

Reading and writing blocks from S.C.M.* is implemented using a look up table stored in LFO* locations 400-480. This scheme takes advantage of the fact that 16 pixels will be taken from each line of the image. It also assumes that the image is made up entirely of 16 x 16 blocks of pixels i.e. each block's starting pixel (number 1) will be stored in the high order byte of bank C in S.C.M. So to read the 16 pixels in the first line of a block whose starting pixel is at address n in bank C:

read the 4 pixels in	n	bank C
	n+1	bank C
then read the 4 pixels in	n	bank D
	n+1	bank D
then read the 4 pixels in	n	bank E
	n+1	bank E
then read the 4 pixels in	n	bank F
	n+1	bank F

Now reset n to the address of the first pixel in line 2 and iterate until 16 lines have been read. (The reason the address is incremented before the bank is that this increment takes place automatically at each DSA* read.) A table showing MAPS order is shown in figure B-5.

*SCM — Scan Converter Memory
LFO — Large file zero
DSA — Direct Storage Access

as they appear on the screen

0	1	4	5	10	11	14	15	40	41	44	45	50	51	54	55
2	3	6	7	12	13	16	17	42	43	46	47	52	53	56	57
8	9	C	D	18	19	1C	1D	48	49	4C	4D	58	59	5C	5D
A	B	E	F	1A	1B	1E	1F	4A	4B	4E	4F	5A	5B	5E	5F
20	21	24	25	30	31	34	35	60	61	64	65	70	71	74	76
22	23	26	27	32	33	36	37	62	63	66	67	72	73	76	77
28	29	2C	2D	38	39	3C	3D	68	69	6C	6D	78	79	7C	7D
2A	2B	2E	2F	3A	3B	3E	3F	6A	6B	6E	6F	7A	7B	7E	7F
80	81	84	85	90	91	94	95	C0	C1	C4	C5	D0	D1	D4	D5
82	83	86	87	92	93	96	97	C2	C3	C6	C7	D2	D3	D6	D7
88	89	8C	8D	98	99	9C	9D	C8	C9	CC	CD	D8	D9	DC	DD
8A	8B	8E	8F	9A	9B	9E	9F	CA	CB	CE	CF	DA	DB	DE	DF
A0	A1	A4	A5	B0	B1	B4	B5	E0	E1	E4	E5	F0	F1	F4	F5
A2	A3	A6	A7	B2	B3	B6	B7	E2	E3	E6	E7	F2	F3	F6	F7
A8	A9	AC	AD	B8	B9	BC	BD	E8	E9	EC	ED	F8	F9	FC	FD
AA	AB	AE	AF	BA	BB	BE	BF	EA	EB	EE	EF	FA	FB	FE	FF

each of the rows is read
in the following order

1	3	5	7	2	4	6	8
bank C	D	E	F	bankC	D	E	F

2 pixels/word

Figure B-5. Derivation of MAPS Ordering Look-Up Table

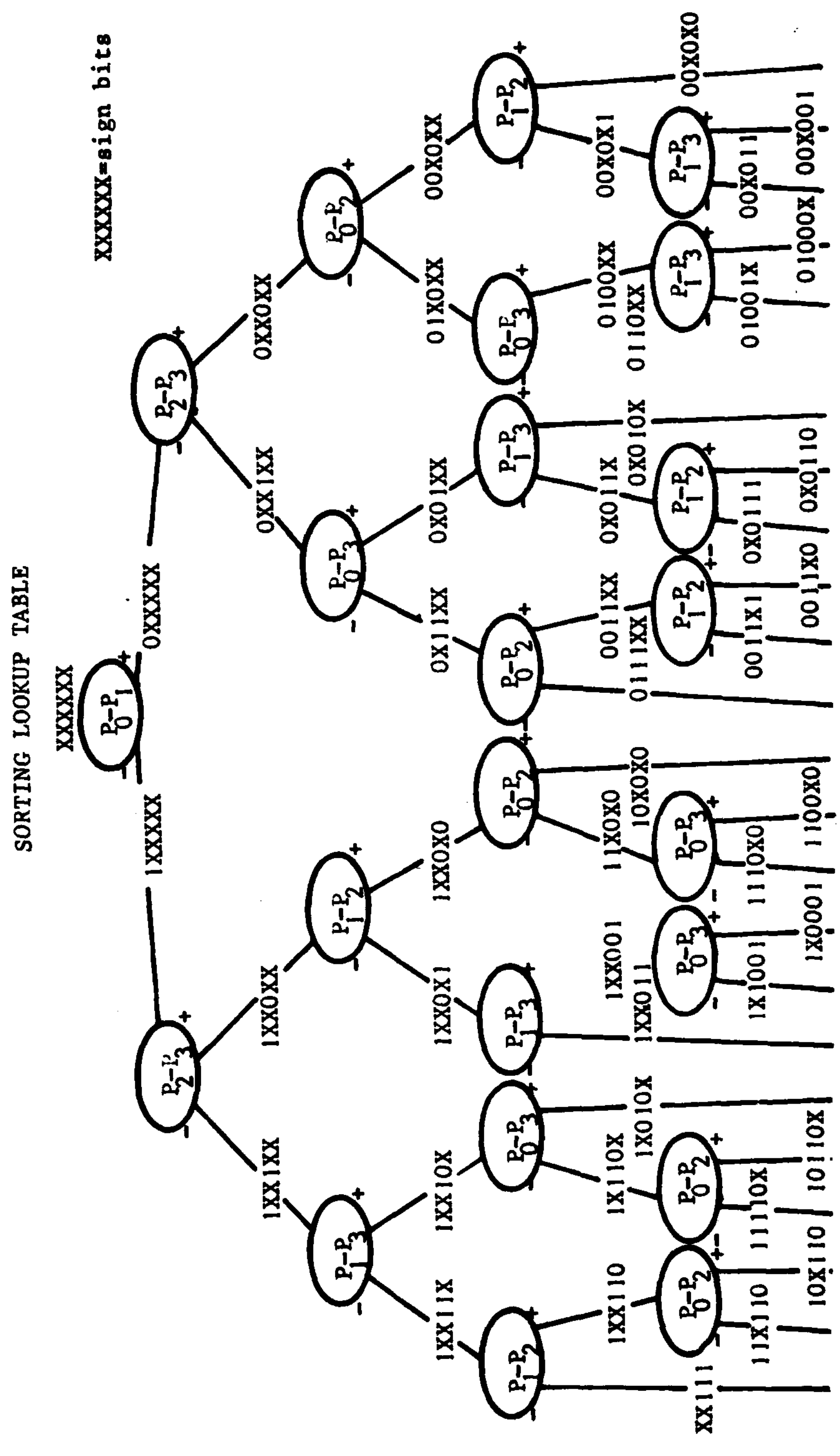
B.3 Sorting Lookup Table

The sort of 4 pixels performed in COMB is achieved using a lookup table. The 4 pixels are located in TF1* locations 0-3. All six possible differences are taken and stored in TFO* locations 0-5. The differences are taken in the following order:

$$P_o-P_1$$
$$P_o-P_2$$
$$P_o-P_3$$
$$P_2-P_3$$
$$P_1-P_3$$
$$P_1-P_2$$

The reason this order was chosen is that this order minimizes the number of changes to the input registers of the adder. The sign bit of each difference is shifted into a register forming a 6 bit address. This address is used to look up the addresses of the extreme, middle, low, and high steps. Figure B-6 shows how the look up table is generated. The table occupies 64 locations in LFO* but only 24 are used. (The Ad column of the figure shows the offset into the table of the required addresses shown in column HMLE. The column marked ORDER shows the actual order of the pixels.)

*TFO — Temporary file zero
TF1 — Temporary file one
LFO — Large file zero



Ad	HMLE	ORDER
00	0532	0123
01	1542	0213
03	1340	0231
11	1043	2013
13	1245	2031
1B	3205	2301
04	0431	0132
06	2451	0312
07	2350	0321
0E	2053	3012
0F	2154	3021
1F	3104	3201
20	0134	1023
30	5124	1203
38	5320	1230
31	5023	2103
39	5421	2130
3B	3401	2310
24	0235	1032
2C	4215	1302
3C	4310	1320
2E	4013	3102
3E	4512	3120
3F	3502	3210

Figure B-6. Derivation of Sorting Look Up Table

TABLE B-1

Glossary

ARDF	-	(MAPS)	Area definition routine
ASCI	-	(LF1)	TO values stored in ASCII
BASE	-	(LF1)	Base Contrast Control Matrix
BCNT	-	(LF1)	Block count (number of compressed pixels)
CCMS	-	(LFO)	Table of 4 CCMS
CGO1	-	(DCON)	Generates one character
CLR	-	(DCON)	Clears a line of characters
CLRI	-	(DCON)	Clears an image
COF1	-	(DCON)	Turns cursor off
COMB	-	(MAPS)	Compresses a block of pixels
CON1	-	(DCON)	Turns cursor on
DBUF	-	(DCON)	Dumps buffer of characters
DECB	-	(MAPS)	Decompresses a block of pixels
DECI	-	(MAPS)	Decodes an integer in ASCII
DRAW	-	(DCON)	Draws lines
DV32	-	(MAPS)	32 bit divide
ENCF	-	(MAPS)	Encode a fixed point number into ASCII
FRAM	-	(LF1)	Line descripter table for DRAW
FUN	-	(MAPS)	Macro Fidelity executive routine
HSCG	-	(LF1)	DCON's CGPT
INTO	-	(MAPS)	Accepts TO input from LSC keyboard
MAPO	-	(LFO)	MAPS ordering lookup table
MAPS	-	(MAPS)	Compression and decompression executive
MES1	-	(LFO)	"TO (0) = 00"

TABLE B-1

(continued)

MES2	-	(LFO)	"INPUT TO VALUE"
MES3	-	(LFO)	"HOLD, GO, MARK, SEND"
MES4	-	(LFO)	"C. R. = "
MES5	-	(LFO)	"CLEAR AREAS?"
MSP	-	(DCON)	16 x 16 bit multiply
MYCG	-	(LF1)	MAPS'S CGPT
OCR	-	(MAPS)	Output compression ratio
OUTT	-	(MAPS)	Output table of TO values
PNTE	-	(MAPS)	Display ">" at table entry
RBUF	-	(MAPS)	Read character from buffer
REDB	-	(MAPS)	Reads a block of pixels from SCM
SORT	-	(LFO)	Sorting lookup table
STAT	-	(LFO)	Area definition state table
TOEQ	-	(MAPS)	Output single TO value
WAIT	-	(MAPS)	Wait for keyboard input
WBUF	-	(DCON)	Write character into buffer
WRTB	-	(MAPS)	Write block of pixels

Appendix C

Selected Microcode Source

Shown here is the source microcode which perform MAPS compression and decompression. Included are MAPS, COMB, REDB, DECB, and WRTB.

```
MPAS2 - VERSION 2.3.5          MAP                  PAGE  16                      DATE - 03/21/86

                          *
                          *     MAPS  EXECUTIVE  (ECON BACKGROUND ROUTINE)
                          *
09D9  4200 0039 3500      MAPS TC      CL2                 $0009         TCHA
09DA  4300 7C00 1541           TC      CL3                 $7C00         TFOC        F1
09DB  4000 0400 1500           TC                          $0400         TFOC
09DC  0000 0400 1500           TR                                  F1    TFCC
09DD  4000 0400 1500           TC                          $0400         TFOC
09DE  4000 001D 000F           TC                          $001D                     ICR2
09DF  4020 0400 1000           TC           SK             $0400         TFOC
09E0  4000 001F 007F           TC                          $001F                     ICR3
09E1  4000 0006 2000           TC                          $0006         TCRA
09E2  0000 3C7A 6E41           TR                    ZRO           A0    LOAC  TFGC  F1
09E3  0000 0330 2E00           TR                                  LFON  LOAC
09E4  0000 0700 0000           TR                                  LFCU  ACF
09E5  4000 000E 1630           TC                          $0006         LFOC        CMR3
09E6  4020 0A80 3800           TC           SK           , RFDB          MAR
09E7  4000 0040 2000           TC                          $0040         FO
09E8  0000 0030 0000           TR
09E9  0000 F072 48A1           TR                    E             LF1D  EO    A1    GO
09EA  0000 8771 48AC           TR                    AND           LF1U  EO    AC    LF1D
09EB  0000 6031 400C           TR                    AND                       AO    LF1N
09EC  0000 0100 0000           SH                                  RZINN A     N
09ED  0000 7C40 C800           TR                    NL7           A0    EO
09EE  0000 0100 0000           SH                                  RZINN A     N
09EF  4000 04C3 0021           TC                          CONS                      GO
09F0  4000 0530 005C           TC                          $0530                     L1AD
09F1  0000 9031 000C           TR                    ADD                       AO    LF1N
09F2  4000 0040 0000           TC                          $0040         LFON        CMRD
09F3  4020 0A0F 3800           TC           SK             CONB          MAR
09F4  0000 0030 0000           TR
09F5  4020 0A9E 3800           TC           SK             DFCB          MAR
09F6  0000 0030 0000           TR
09F7  4000 0007 2000           TC                          $0007         TORA
09F8  0000 0E3C 2000           TR                                  TFOC  FO    LFGU  NOP
09F9  4000 004E 0021           TC                          $0040                     G8
09FA  4020 0AC3 3800           TC           SK             WRTB          MAR
09FB  4000 0006 1630           TC                          $0006         LFCC        CMR3
09FC  4000 000E 3C00           TC                          $0006         TOEA
09FD  0000 0730 0000           TR                                  LFOU  ACF
09FE  4000 0040 0630           TC                          $0040         LFGN        CMR3
09FF  0000 001A 0021           TR                                              TFOU  G8
0A00  4000 0002 0800           TC                          $0002         EO
0A01  1916 9C5A 0021           TR AD   IN3 JP        ADD           A0    TFOL  TFGU  GO
0A02  0000 9C40 8000           TR                    ADD           A0    TFCL
0A03  4000 000E 3000           TC                          $0008       , TOEA
0A04  4000 0010 1630           TC                          $0010         LFCC        CMR3
0A05  0300 001A 0021           TR      CL3                                     TFOU  GO
0A06  4000 0400 0800           TC                          $0400         EC
0A07  0000 9C40 8000           TR                    ADD           A0    TFOL
0A08  0000 9C7A A0A1           TR                    ADD           A0    FC    TFGC  G8
0A09  0000 9C71 8521           TR                    ADD           A0    TFGN  A0    GO
0A0A  4000 000E 3500           TC                          $0006         TCHA
```

```
        MPAS2 - VERSION 2.3.5        MAPS            PAGE  17                DATE - 03/21/80

GAGB  180C 000C 000C          TP AC  II2 JF                 PG    TFCC                                618
GAGC  000C AC00 4500          TR                G           AG    TFGN                                619
GAGD  461C 0001 0E3D          TC            JF        $0001         LF0C      CMR3                    620
GAGE  4600 003C 104F          TC            IF        $0000         LFGC      ICR0                    621
```

```
                                 *
                                 *     MAPS - COMPRESSION MODULE      FOR USE AS  CCCN BACKGROUND ROUTINE
                                 *
                                 *       REQUIREMENTS  (THESE REQUIREMENTS ARE ALSO MET ON EXIT)
                                 *         LC/D = CMRG SAVE LOCATION IN INTERRUPT SAVE STACK
                                 *         CM+G = $0046
                                 *         CM+3 = $000C   (AD CONDITION MUST BE FALSE ON ENTRY)
                                 *         LC KUF TABLE FOR 4 PIXEL SORT IN  LF6481 - LF64C0
                                 *         CCN POINTEC TO BY (LF1530)
                                 *         LOCKUP TABLE CONVERTING LEVEL TO PIXEL SIZE IN LF152A- LF152F
                                 *
                                 *       STORAGE USED:                         *   INPUTS
                                 *         TF 0 - TF05            F0  F1       *     256 5 BIT PIXELS IN LF4CC- LF4FF
                                 *         TF10 - TF08            G0  G1       *      (IN MAPS ORDER)
                                 *         LF 4C1-2               PAST         *
                                 *         E0  E1                 IDXC         *   OUTPUTS
                                 *         ICF0                                *     MAPS PIXELS IN LF1400-LF14FF
                                 *                                             *      (IN EXPANDED FORMAT)
0A0F  C28C 0037 0021     COMB TR     C11                   ----------             LOAD  G0  ---------------
0A10  4000 0530 005C          TC                   $0530                                L1AD
0A11  0000 0340 2E00          TR                                      LF1N  LOAC               INITIALIZE
0A12  4000 0034 00EA          TC                   $0044                                T1NA
0A13  0000 3C5C 751A          TR              ZRO                     AO    TONA  LF0U  TF1U   LOAD CCM
0A14  0000 0E9C 001A          TR                                      TF0C  NCF   LF0U  TF1U
0A15  0000 AC5C 509A          TR              G                       A0    G1    LF0U  TF1U
0A16  0000 001C 001A          TR                                                  LF0U  TF1U
0A17  0240 0037 000C          TR     CL0                                          LOAD  LF1N
0A18  4000 052E 005C          TC     IF0           $052E                                L1AD
0A19  4000 04C1 2E00          TC                   $04C1                    LOAC
0A1A  0000 002C 005F          TR                                                  LF1D  ICR1
0A1B  0000 30F1 463A          TR              ZRO                     L1AD  LF0N  A0    TF1C
0A1C  0000 AC80 6E00          TR              G                       A1    LOAC
0A1D  0000 3731 4021          TR              ZRO                     LF0U        A0    G0
0A1E  0000 0700 0000          TR                                      LF0U
0A1F  4000 0430 005C          TC                   $0400                                L1AD
0A20  4000 0041 0E4F          TC          SK       $0041                    LFCN        ICR0
0A21  0000 AC80 0E00          TR              G             ---------A1     LOAD              ---------------
0A22  0000 0F7C 089A          TR     IN1                      P0      LF1A  E0    LF1A  TF1U   GET NEXT 4
0A23  0800 6F40 0000          TR IN           SUB                     LF1A  NCF
0A24  1000 6F7C 089A          TR FN           SUB             P1      LF1A  E0    LF1A  TF1U   PIXELS. ARE
0A25  0800 6F4C 0000          TR IN           SUB                     LF1A  NCF
0A26  1000 6F7C 089A          TR FN           SUB             P2      LF1A  E0    LF1A  TF1U   THEY ALL THE
0A27  1000 634C 08EA          TR FN           SUB             P3      LF1N  E0    LF1N  TF1C   SAME LEVEL
0A28  4000 0000 307A          TC                   $0000                    TOEA        T1BA
0A29  C400 E03E 0000          TR UNS          SUB                                 TF1C  NOP
0A2A  E010 0A05 3800          TC TF       JF       P2       ----------      MAR               ---------------
0A2B  0000 0EC0 0800          TR                                      TF1U  E0                 CALCULATE
0A2C  0000 3C5E 00A1          TR              ZRO                     A0    F0    TF1U  G0       THE 6
0A2D  8000 6002 2000          SM              SUB                           LA15N               POSSIBLE
0A2E  0000 EC5E 0021          TR              SUB             P0-P1   A0    TFCL  TF1U  G0     DIFFERENCES.
0A2F  8000 0002 2000          SM              SUB                           LA15N              STORE THEM
0A30  0000 EC6E 0021          TR              SUB             P0-P2   A0    TFCL  TF1D  G0      IN TFC0 -
0A31  8000 6002 2000          SM              SUB                           LA15N                TF05.
```

```
0A32  0000 EC4C 0000          TR              SUB        P0-P3   AG     TFCU                    SHIFT THE
0A33  0000 CAC0 080C          TR                                 TF1D   EC                      SIGN BIT OF
0A34  6000 6032 2000          SH              SUB                       LA1SN                   EACH INTO
0A35  0000 EC40 0D00          TR              SUB        P2-P3   AG     TFCL                       F0.
0A36  0000 0EC0 0880          TR                                 TF1U   E0
0A37  6000 6032 2000          SH              SUB                       LA1SN
0A38  0000 EC58 0021          TR              SUB        P1-P3   AG     TFCU   TF1U   G0
0A39  8000 6002 2000          SH              SUB                       LA1SN                   USING F0
0A3A  0000 EC43 1021          TR              SUB        P1-P2   A0     TFCC   F0     G0      LOOK UP THE
0A3B  4000 0010 0000          TC                   $0010                LF0N          CMR0      LOCATION OF
0A3C  4000 0481 0880          TC                   $0481                EC                        E M H L
0A3D  0000 9C40 AE00          TR              ADD                AG     LOAC                       STEPS
0A3E  0000 000C 0011          TR                         ---------             LFGN   E1      --------------
0A3F  0000 FC9E 6000          TR              E                  A1     TORA   TF1U   NOP
0A40  0000 06FA 08A1          TR                         EXTREME TF1U   E0     TFGC   G0      COMPARE
0A41  0000 9031 8021          TR TNS          ADD         STEP                 AG     G0      DIFFERENCES
0A42  3000 6031 8021          TR FF           SUB                              A0     G0      WITH C.C.P.
0A43  4C10 0A61 3880          TC TNS     JF        H1                   PAR
0A44  0000 F080 6000          TR              E          ---------A1SW  TORA
0A45  0000 06FA 08A1          TR                                 TF1U   E0     TF0C   G0
0A46  0000 9031 8021          TR TNS          ADD        MIDDLE                A0     G0
0A47  3000 6031 8021          TR FP           SUB         STEP                 A0     G0
0A48  4C10 0A61 3880          TC TNS     JF        H1                   PAR
0A49  8000 0642 1000          SH                         ---------LCIRN LCIRN         D
0A4A  8000 0642 1600          SH                                  LCIRN LCIRN         D
0A4B  8000 0642 1000          SH                                  LCIRN LCIRN         D
0A4C  8000 0642 1000          SH                                  LCIRN LCIRN         D
0A4D  0000 F080 6000          TR              E                  A1SW   TORA
0A4E  0000 06FA 08A1          TR                                 TF1U   E0     TF0C   G0
0A4F  0000 9031 8021          TR TNS          ADD        LOW                   A0     G0
0A50  3000 6031 0021          TR FF           SUB        STEP                  A0     G0
0A51  4C10 0A61 3880          TC TNS     JF        H1                   PAR
0A52  0000 80C6 2000          TR                         ---------F0    TCRA
0A53  0000 0EFA 08A1          TR                                 TF1C   E0     TFGC   G0
0A54  0000 9031 8021          TR TNS          ADD        HIGH                  AG     G0
0A55  3000 0631 0021          TR FP           SUB        STEP                  A0     G0
0A56  4C00 0A61 3880          TC TNS               H1                   PAR
0A57  0000 36F1 4891          TR              ZRO        ---------TF1U  E0     A0     E1      --------------
0A58  0000 081E 8021          TR                          P0                   TF1U   G0      CALCULATE
0A59  0000 9C5E 08A1          TR              ADD        +P1     AG     EG     TF1U   G0      NEW INTENSE-
0A5A  0000 9C5E 08A1          TR              ADD        +P2     AG     E0     TF1U   G0      TY AND
0A5B  0000 9C40 8880          TR              ADD        +P3     AG     E0                      RESOLUTION
0A5C  4000 0092 0021          TC                   $0082 + 82/                        G0
0A5D  0000 9C40 8080          TR              ADD          /4    AG     E0
0A5E  8000 013C 8000          SH                            /    RZINN
0A5F  8000 0100 8000          SH                                 RZINN
0A60  0000 F031 400C          TR              E                                A0     LF1N
0A61  0000 AC8C 6E00     H1   TR              G          ---------A1    LOAC                  --------------
0A62  4000 0848 0600          TC                   $0040                LFCA          CMR0    LEVEL
0A63  4000 0008 005A          TC                   $0008                              T1RA
0A64  0000 683E 0021          TR                                               TF1C   G0      COMPLETE
0A65  1016 083E 0000     H2   TR AC      JF                                    TF1C   NOP     --------------
0A66  0000 0DFC 2000          TR                                 L1AD   F0     LF1A   NOP
```

```
MPAS2 - VERSION 2.3.5          MAPS               PAGE  20                   DATE - 03/21/80

0A67  0040 00C0 0080          TR    I/O                      L1AD  EO               UCAE
0A68  0000 AC43 60A1          TR               G             AG    FG    FO    GO
0A69  5010 0A7C 3600          TC AO       JF          OUT          MAR
0A6A  0000 6C40 0080          TR               SUB           AG    EO
0A6B  0000 CC40 0080          TR               E+E           AG    EO
0A6C  0000 C631 800F          TR               E+E                       AO    IOXG
0A6D  4000 0530 005C          TC                      $0530                    L1AD  GET NEXT
0A6E  0000 034C 2E21          TR                             LF1N  LOAC  LF1N  GO    C.C.M.
0A6F  4280 0C04 00LA          TC    CL1               $C004        EO          T1MA  RESET FIXEL
0A70  0000 9031 800C          TR               ADD                       AO    LF1N  INCREMENT.
0A71  0000 001C 001A          TR                                         LFGU  TF1U -RESET LOOF
0A72  0000 001C 001A          TR                                         LFGU  TF1U  COUNTERS.
0A73  4000 0028 0021          TC                      $0020                    GO    UPDATE LEVEL
0A74  C000 00DC 085A          TR                             RO    EO    LFGU  TF1U  TESTER.
0A75  0000 001C 001A          TR                                         LFGU  TF1U
0A76  0000 9031 803A          TR               ADD                       AC    TF1C
0A77  4C00 04C1 2E00          TC                      $04C1        LOAC
0A78  0C00 003C 005C          TR                                         LFON  L1AD
0A79  0000 002C 005F          TR                                         LF1D  ICR1
0A7A  0C00 90F1 0621          TR               ADD           L1AD  LFUA  AG    GO
0A7B  4C10 03FF 005C          TC          JF          $03FF                    L1AD
0A7C  0600 AC0C 6E00          TR               G             A1    LOAD  LF1A  NOP   --------------
0A7D  0030 063C 0600    OUT   TR          CF
0A7E  0016 0000 0600          TR           F
0A7F  0030 0000 0060          TR          CF
```

```
                                *
                                *   MAPS READ BLOCK MODULE    FOR USE AS DECON BACKGROUND ROUTINE
                                *
                                *      REQUIREMENTS   (THESE ARE ALSO MET ON EXIT)
                                *        L0/D = CMRC SAVE LOCATION
                                *        CM:3 = $0006
                                *        LOOKUP TABLE FOR MAPS ORDERING IN LF0400-LF0480
                                *
                                *      STORAGES            *    INPUTS
                                *        E0 E1   ICR0      *      F0 = (YMAX+7)/8
                                *        F0 F1   IOX1      *      F1 = ADDRESS OF STARTING PIXEL
                                *        G0 G1   ICR1      *    OUTPUTS
                                *        ICX0              *      PIXELS IN LF1400-4FF (IN MAPS ORDER)
                                *                          *
                                *                          *
0A80  4000 000C 0880            RECB TC                    $C00C          E0
0A81  4240 0004 005F                 TC      C10           $0064                      ICR1
0A82  C280 C730 C00C                 TR      C11                     LFCU  ACF
0A83  0000 0730 0000                 TR                              LFOU  ACF
0A84  4000 C010 004F                 TC                    $0010           LFCN       ICR0
0A85  4C00 0400 2C00                 TC                    $0400           LOAC
0A86  4C20 1F1F 0011                 TC         SK         $1F1F                      E1
0A87  0020 001C 005C                 TR         SK                              LFOU  L1AD
0A88  C03C F4F1 6C00                 IO         CF   E               F1         A0    DS   NOPNOPNCPLSB
0A89  CCA0 C000 2800                 IO      IN1 SP                                   DS   NOPNOPNCPR
0A8A  0C30 BC32 50A1                 TR         CF   AND             A1    G1   A1    G0
0A8B  0020 A842 509C                 TR         SP   G               AGRZ  G1   A1SH  LF1U
0A8C  0030 A022 400C                 TR         CF   G                          A1RZ  LF1N
0A8D  C020 0C5C 080C                 TR         SF   E+1             A0    E0   LFOU  L1AD
0A8E  0030 0040 1080                 TR         IF                   23    C1
0A8F  C020 0030 2800                 IO         SF                                    DS   NOPNOPNOPR
0A90  0030 BC02 5CA1                 TR         IF   AND             A1    C1   A1    G0
0A91  0020 A842 509C                 TR         SP   G               AORZ  G1   A1SH  LF1U
0A92  0030 A022 400C                 TR         IF   G                          A1RZ  LF1N
0A93  1016 001C 005C                 TR AD      JP                              LFCU  L1AD
0A94  0020 0D4C 1080                 TR         SF                   23    C1
0A95  0070 04C3 08A1                 TR      ING IF                  F1    E0   F0    G0
0A96  0000 9031 8041                 TR              ADD                        A0    F1
0A97  5016 0A3E 3880                 TC AC      JP         CCNE            PAR
0A98  0200 A032 4021                 TR      C11     G                          A1    G0
0A99  4016 000C 0080                 TC         JF         $000C           E0
0A9A  0020 001C 005C                 TR         SK                              LFOU  L1AC
0A9B  0030 3C4C 0E5C            CONE TR         IF   ZRO             A0    LCAC LFON  L1AD
0A9C  0016 A842 509C                 TR         JP   G               AORZ  C1   A1SH  LF1U
0A9D  0036 A322 0E0C                 TR         LF   G               LFCN  LOAC A1RZ  LF1N
```

```
                            *
                            *  MAPS -  DECOMPRESSION MODULE      FOR USE AS  DCON BACKGROUND ROUTINE
                            *
                            *    REQUIREMENTS  (THESE REQUIREMENT ARE MET ON EXIT)
                            *      L0 D   CMR0 SAVE LOCATION IN INTERRUPT SAVE STACK
                            *      CM+0 = $004C
                            *      CM 3 = $0006 (A0 CONDITION MUST BE FALSE ON ENTRY)
                            *      LOOKUP TABLE CONVERTING LEVEL TO PIXEL SIZE IN LF152B-LF152F
                            *
                            *    STORAGE USED:                *  INPUTS
                            *      TF 0 - TF04     PAST       *    MAPS PIXELS IN LF400- LF14FF
                            *      E0 E1           ICR1       *     (IN EXPANDED FORMAT)
                            *      F0              ICR0       *  OUTPUTS
                            *      ICX0            ICR1       *    256 5 BIT PIXELS IN LF1400-LF14FF
                            *                                 *     (IN MAPS CODEF DECOMPRESSED)
                            *                                 *    PIXEL COUNT AT BCNT IS
                            *                                 *     AUGMENTED
                            *                                 *                                  --------------
0A9E  4000 0528 005C    DECB TC                     $0528                           L1A0  INITIALIZE
0A9F  4000 00C0 3500         TC                     $0000             TCRA
0AA0  .000 075C 0000         TR                               LF10   TF0L  LF00  NOP
0AA1  .000 075C 0000         TR                               LF10   TF0L  LF00  NOP
0AA2  4000 0130 164F         TC                     $0100             LF0C         ICR0
0AA3  0000 0740 0000         TR                               LF10   TF0L
0AA4  0240 0740 0000         TR     CL0                       LF10   TF00
0AA5  .000 0340 0000         TR                               LF1N   TF0L
0AA6  4000 04FF 005C         TC                     $04FF                           L1A0
0AA7  .020 3C00 6080         TR           SR     ZR0          A0     F0
0AA8  4000 0020 0021         TC                     $0020                           G0    --------------
0AA9  5010 0A3C 3000         TC A0        JF     H3                  MAR                   CORE
0AAA  0040 0B6C 0051         TR     IN0                       LF10   E0    LF10  E1
0AAB  0400 0000 0680         TR UNS              SUB          F0     E0                    ZERO LEVEL
0AAC  7010 0A33 3000         TC FF        JF     BIGP                MAR
0AAD  .040 0C5C 2800         TR     IN0          E+1          A0     E0FC  LF10  NOP  YES, THEN
0AAE  .040 0C5C 0680         TR     I10          E+1          A0     E0    LF10  NOP  SKIP 3 PIXELS
0AAF  .050 0C0C 0680         TR     I10 JF       F+1          A0     E0    LF10  NOP
0AB0  .000 0C40 2000         TR                  E+1          A0     E0FC
0AB1  0140 F032 4041    BIGP TR     D10          E                         A1    F1   NOP
0AB2  .000 0032 8011         TR                  F+E                       A1    E1
0AB3  0280 0032 8011         TR     C11          E+E                       A1    E1   EXTRACT
0AB4  0000 C03C A000         TR     I11          E+E          A1SW   TCRA  LF10  NOP  RESOLUTION
0AB5  4020 0034 003D         TC           SR     $0004               LF0N         CMR3
0AB6  .000 003A 005F         TR                                            TF0N  ICR1  EQUATE APPRO-
0AB7  1050 .013 002C         TR A0  IN1 JF                                 F1    LF10   PRIATE NUMBE
0AB8  .040 0030 0000         TR     I10                                              R OF PIXELS.
0AB9  4170 003E 003C         TC     C 0 CC       $0000               LF0N         CMR3
0ABA  .010 003C 0000         TR         JF
0ABB  .000 0030 0000         TR
0ABC  0000 0000 000C         TR
0ABD  4000 0529 005C    H3   TC                  BCNT                              L1A0
0ABE  0000 3772 5091         TR              ZR0              LF10   C1    A1    E1
0ABF  .030 00CE 00A1         TR           CF                  F0     E0    LF1N  G0   UPDATE BLOCK
0AC0  .010 9831 A02C         TR           JF     ADD          LF.D   NCF   A0    LF10 COUNTER.
```

```
        MPAS2 - VERSION 2.3.5        MAPS          PAGE  23             DATE - 03/21/80

GAC1  0030 5032 8000          TR       IF      ADD                     A1    LFIN                    854
BAC2  0000 0030 0000          TR                                                                     855
```

```
                         *
                         *     MAPS - WRITE BLOCK ROUTINE     FOR USE AS ECCN BACKGROUND ROUTINE
                         *
                         *        REQUIREMENTS (THESE ARE MET ON EXIT)
                         *          L0 D = CHRC SAVE LOCATION
                         *          CM13 = $L0GE
                         *          LOOKUP TABLE FOR MAPS ORDERING IN LF040C-480
                         *
                         *        STORAGES                    *    INPUTS
                         *          E0 E1    ICR1             *      F0 = ADDRESS OF STARTING PIXEL
                         *          F0 F1                     *      G0 = (YMAX+7)/8
                         *          G0 G1                     *    OUTPUTS
                         *          ICX0                      *      PIXELS IN SCM
                         *          IC(0                      *
                         *          IC)0                      *
                         *
0AC3  4000 000C 0880     WRTB TC                        $000C BANK 0      E0
0AC4  4200 0004 005F          TC     C11                $0004                         ICR1
0AC5  0240 0737 0041          TR     C10                           LF0U  NCF   LEAD  F1
0AC6  0000 070C 0000          TR                                   LF0U  NCF
0AC7  4000 0010 004F          TC                        $0010            LFCN        ICRC
0AC8  4020 0431 2000          TC        SR              $0401            LOAC
0AC9  0020 001C 005C          TR        SR                                     LF0U  L1AC
0ACA  0080 07-0 1080          TR     IN1 DF                        LF1U  C1
0ACB  0020 A342 5091          TR        SF    G                    LF1N  C1    A1SH  E1
0ACC  0030 001C 005C          TR        CF                                     LF0U  L1AD
0ACD  0020 F0F1 0000          IO        SR    E                    R0          A0    DS    NCFNOPNCPLSR
0ACE  C030 E032 6A00          IO        CF    OR                               A1    DS    NOPNOPNCPN
0ACF  0026 0740 1C80          TR        SR                         LF1U  C1
0AD0  0030 A342 5051          TR        IF    G                    LF1N  C1    A1SH  E1
0AD1  1010 0C5C 00CC          TR AC     JF    E+1                  A0    E0    LF0U  L1AD
0AD2  C020 E032 6A00          IO        SR    OR                               A1    DS    NCFNCFNCFN
0AD3  0070 000C 00CC          TR     IN0 EF                        F0    E0    LF0U  L1AD
0AD4  C000 9C40 A080          TR              ADD                  A0    F0
0AD5  5E10 0A05 3800          TC AG     JF          SFVA                 MAR
0AD6  1030 C030 0000          TR AD     IF
0AD7  4250 000C 0880          TC     CL1 JF             $000C            E0
0AD8  0020 000C 0000          TR        SR
0AD9  0010 0400 2E00     SEVA TR        JF                         F1    LOAC
0ADA  0030 003C 0000          TR        EF
                         *END
```